DOCUMENT RESUME

ED 108 613 IR 002 128

AUTHOR Kline, Lanaii
TITLE CONC-Dictionary.
INSTITUTION Southwest Regional Laboratory for Educational Research and Development, Los Alamitos, Calif.
REPORT NO SWRL-TN-5-72-10
PUB DATE 29 Feb 72
NOTE 80p.

EDRS PRICE MF-$0.76 HC-$4.43 PLUS POSTAGE
DESCRIPTORS *Computer Programs; Data Processing; *Dictionaries; Essays; *Information Storage; Input Output Devices; Vocabulary; Word Frequency; *Word Lists
IDENTIFIERS *CONC Dictionary

ABSTRACT

The computer program sets up three dictionaries: basic, fiction, and fantasy. It then processes each essay by parsing the essay into words, checking if the word is a legal word (in the dictionary), and adding it to the essay unique word list. If the word was non-unique, the frequency count for that word in the essay word list is increased by one. When the program reaches the end of the essay, two reports are generated, words ordered alphabetically and words ordered by their frequency. The report gives complete documentation and includes a description of subroutines and subfunctions, data specifications, operating instructions, and flowchart layout forms. (Author/SK)

****
* Documents acquired by ERIC include many informal unpublished *
* materials not available from other sources. ERIC makes every effort *
* to obtain the best copy available. nevertheless, items of marginal *
* reproducibility are often encountered and this affects the quality *
* of the microfiche and hardcopy reproductions ERIC makes available *
* via the ERIC Document Reproduction Service (EDRS). EDRS is not *
* responsible for the quality of the original document. Reproductions *
* supplied by EDRS are the best that can be made from the original. *
****

ED108613

SWRL

SOUTHWEST REGIONAL LABORATORY
TECHNICAL NOTE

DATE: February 29, 1972

NO: TN 5-72-10

TITLE: CONC-DICTIONARY

AUTHOR: Lanaii Kline

ABSTRACT

The program sets up three dictionaries; basic, fiction, and fantasy words. It then processes each essay by parsing the essay into words, checking if the word is a legal word (in the dictionary), and adding it to the essay unique-word list.

U S DEPARTMENT OF HEALTH, EDUCATION & WELFARE
NATIONAL INSTITUTE OF EDUCATION
THIS DOCUMENT HAS BEEN REPRODUCED EXACTLY AS RECEIVED FROM THE PERSON OR ORGANIZATION ORIGINATING IT. POINTS OF VIEW OR OPINIONS STATED DO NOT NECESSARILY REPRESENT OFFICIAL NATIONAL INSTITUTE OF EDUCATION POSITION OR POLICY

This document is intended for internal staff distribution and use. Permission to reprint or quote from this working document, wholly or in part, should be obtained from SWRL, 11300 La Cienega Boulevard, Inglewood, California, 90304.

IR 000 128

1.0 - PROGRAM IDENTIFICATION

CONC-DICTIONARY

2.0 - OBJECTIVE

The program sets up three dictionaries; basic, fiction, and fantasy words. It then processes each essay by parsing the essay into words, checking if the word is a legal word (in the dictionary), and adding it to the essay unique-word list.

3.0 - PROGRAM DESCRIPTION

3.1 - Program Logic

The program sets up three dictionaries and then parses each input essay into words. The word is then determined to be a legal word or not. A legal word is determined by its existence in one of the dictionaries: basic and fiction for a fiction essay, basic and fantasy for a fantasy essay, and basic for a transitional essay or animal essay. A word must also appear in its story number range in the dictionary. For example, if the word is in fantasy story number five, then the word must be basic or in fantasy words one through five. If it appears in fantasy six words, it is not a legal word for that essay. If the word is determined to be illegal, then an error message is printed, and the program processes the next word. If the word is legal, the program checks if it is the word, and or but and it succeeds terminating punctuation (a period, question mark, etc.). If it is the word, and or but, and it starts a sentence, then a message is printed, and the program processes the next word; otherwise, the word is packed and determined to be unique or non-unique. If it is unique, the word with a frequency count of one is added to the essay word list, and the program processes another word. If the word was non-unique the frequency count for that word in the essay word list is increased by one. The program then processes another word. When the program reaches the end of the essay, two reports are generated, words ordered alphabetically and words ordered by their frequency. The program continues until all the essays have been processed.

3.2 - Variables and Internal Tables

Integers:

BLK - is the alpha character ' '.
STAR - is the alpha character '*'.

ENDLST - contains the alpha characters 'END '.
FICT - contains the alpha characters 'FIC '.
FANT - contains the alpha characters 'FAN '.
BASE - contains the alpha characters 'BAS '.
PER - is the alpha character '.'.
QUES - is the alpha character '?'.
EXCLAM - is the alpha character '!'.
NBAS - is the number of words in the array, BASIC.
NFAN - is the number of words in the array, FAN.
NFIC - is the number of words in the array, FIC.
NB - is the number of characters in the current word. NB ≤ 16.
N1 - is a flag. If N1 equals one, then the program is processing the first word on the input card. If N1 equals two, then it is on the second word.
PFLAG - is a flag which indicates that terminating punctuation was found. If PFLAG equals one, then the program processes the last word. If PFLAG equals two, then the program processes the the word following the punctuation.
WFLAG - is a flag which indicates what type of essay is being processed. (WFLAG=0, basic; WFLAG=1, fiction; and WFLAG=2, fantasy)
THING - is the essay input type, e.g., THING equals BAS, FAN, or FIC.
THING1 - is a filler variable. It contains the remaining portion of BASIC, FICTION, or FANTASY.
NUMB - is the input essay story number.
HIT - is a flag generated in PACK which indicates that a word was found (HIT=I) or not (HIT=0).
OVFLOW - is a flag which indicates that the program overflowed the essay word list. When OVFLOW equals one, intermediate results are to follow.
PCOUNT - is a count of the number of lines printed. PCOUNT ≤ 50.
MORE - is a flag which indicates that there is more of the title to be read.
KCARD - is a count of the number of lines read for that essay.
NSAVE - is the number of unique words in the essay.
NWORDS - is the number of words in the essay.
RDCARD - is a flag which indicates that a word continued onto the next input card; and therefore, a new input card should not be read instead use the BUFFER array to obtain more essay words.
DONE - is a counter which gives the number of characters in the array BUFFER.

NCHAR - is a counter. NCHAR ≤ 4.
OUT - is an output variable. OUT equals BAS, FAN, or FIC.
ENDFLG - is a flag which indicates where the end of the legal word list is for that essay.
LCARD -
WD -

Integer Arrays:

WORD(3000,6) - is the array containing the unique essay words.
WORD(N,1) - WORD(N,4): word in 4 bytes
WORD(N,5) : frequency count
WORD(N,6) : line number of essay line where the word first appeared.

NO(2) - is an input variable which contains the story number of the corresponding input word. NO(1) is the first word number and NO(2) is the second word number.

CARD(80) - contains the current essay input line.

TITLE(38) - contains the current input essay title.

BUFFER(200) - contains the current essay input line plus the rest of the characters of a word that continued onto the next input line.

TWORD(4) - contains the value of a packed word.

CHAR(50) - contains the characters of an unpacked word.

TYPE(2) - is an input array which contains the word type of each word on the input card, e.g., TYPE(1) contains the type for the first word on the card. TYPE may be fantasy (FAN), fiction (FIC), or basic (BAS).

LWORD(2,16) - contains the words on a dictionary input card. LWORD(1,1)-LWORD(1,16) is the first word on the card and LWORD(2,1)-LWORD(2,16) is the second.

BASIC(800,4) - is an array which contains the packed words in the basic word list.

FIC(1500,5) - is an array which contains the packed words in the fiction word list.

FAN(800,5) - is an array which contains the packed words in the fantasy word list.

AND(4) - is an array which contains the packed equivalent of the Hollerith text 'AND '.

BUT(4)	- is an array which contains the packed equivalent of the Hollerith text 'BUT '.
TEST1(4)	- is an array which contains the alpha characters 'A', 'N', 'D', and ' '.
TEST2(4)	- is an array which contains the alpha characters 'B', 'U', 'T', and ' '.

4.0 - SUBROUTINES AND FUNCTIONS

SUBROUTINE PACK(CHAR,TWORD,n,HIT)	- This subroutine encodes an n-character array, CHAR, into a four element array, TWORD. The arguments CHAR and n are input arguments. TWORD and HIT are output arguments. If no letters were found in CHAR, PACK returns a value of zero in HIT. The argument n can either be an integer constant or an integer variable, but neither number be greater than sixteen.
SUBROUTINE UNPACK(BUFFER,CHAR)	- This subroutine unpacks the encoded words. BUFFER is the input argument containing the packed word. CHAR is the output argument containing the decoded word.
SUBROUTINE PAGE(PCOUNT)	- This subroutine causes the line printer to go to top of form. PCOUNT is an input and output argument. PCOUNT equals fifty when the subroutine is called, and PCOUNT equals zero when the program flow is returned to the main program.
SUBROUTINE SORT(n,NSAVE)	- This subroutine sorts the packed word list, WORD, based on the input argument n, an integer constant. If n equals 1, the subroutine sorts the words in WORD alphabetically. If n equals five, SORT orders the words in WORD according to their frequency of occurrence (in ascending order). NSAVE is an input argument which indicates the number of words in the array, LWORD

5.0 - DATA SPECIFICATIONS

5.1 - Input Formats
See Attachment 1

5.2 - Output Formats
See Attachment 2

6.0 - PROGRAM CONSTRAINTS AND LIMITATIONS

6.1 - Programming Language
FORTRAN IV

6.2 - Vendor
University of California, Los Angeles
Campus Computing Network

6.3 - Storage Requirements

	Core	I/O Requests
Compilation:	98K	206
Execution:	448K	*

*This figure is dependent on the number of input cards read and output lines printed.

6.4 - Hardware Configuration
IBM 360 Model 91, printer, card reader, and 2314 disk

6.5 - Program Parameters
The maximum number of unique words per essay cannot exceed 3000 words. The maximum number of words in the basic, fiction, and fantasy dictionaries cannot exceed 800, 1500, and 800 words, respectively.

6.6 - Error Messages

<u>Message:</u> PROGRAM TERMINATED DUE TO ERROR ON TYPE FIELD IN THIS WORD LIST INPUT ON CARD input card

Description: Characters other than the following character sequences were read on a dictionary input record: BAS, FIC, or FAN.

Program Action: Program terminated.

<u>Message:</u> EXCEEDED $\left\{\begin{matrix}\text{BAS}\\ \text{FIC}\\ \text{FAN}\end{matrix}\right\}$ ARRAY

Description: The number of entries in the BAS, FIC, or FAN arrays exceeded 800, 1500, or 800, respectively.

Program Action: Program terminates.

Message: {AND or BUT} BEGINS A SENTENCE

Description: The word "and" or "but" begins a sentence.

Program Action: Program eliminates this word from processing and continues to the next word.

Message: THIS WORD DOES NOT APPEAR IN THE BASIC WORD LIST OR IN {BAS, FIC or FAN} WORD LIST word

Description: An essay word was found which did not appear in the proper place in the basic dictionary or in one of the other dictionaries.

Program Action: The program eliminates the word from the processing and continues.

Message: MAX WORD TABLE LENGTH EXCEEDED

INTERMEDIATE RESULTS WILL FOLLOW

Description: The number of unique words found exceeded 3000 words.

Program Action: The program processes the 3000 unique words, then goes back and processes the rest.

7.0 - OPERATING INSTRUCTIONS

Submit in a batch mode with the following control cards:

```
Col 1
//AGB005nn JOB 'mS,jI,R,BIN=xxxx,ID=SWRL','name'
// PASSWORD password
// EXEC FORTGCG
//FORT.SYSIN DD *
      Main Program
            .
            .
            .
      Subroutines
            .
            .
            .
```

```
/*
//GO.FT06F001 DD UNIT=(SYSDA,5),SPACE=(TRK,(100,100),RLSE)
//GO.SYSIN DD *
        Dictionary Input
              .
              .
              .
Col 1
END
        Identification
        Title
        Essay
              .
              .
              .
*       Identification
        Title
        Essay
              .
              .
              .
*
              .
              .
              .
*
/*
//
```

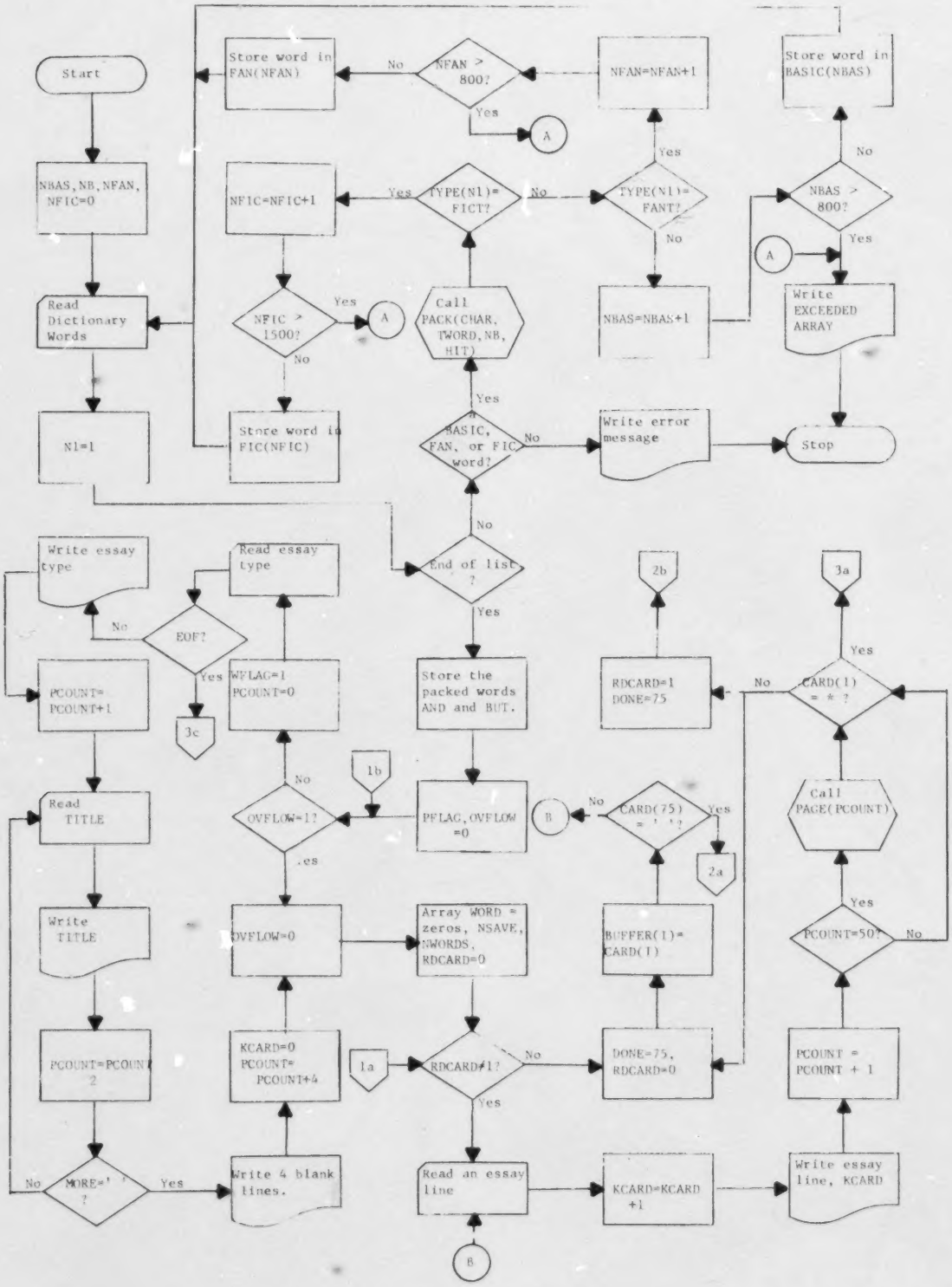
See Attachment 3 for Sample Input and Attachment 4 for Sample Run.

8.0-PROGRAM FLOWCHART

SWRI

FLOWCHART LAYOUT FORM

PROGRAMMER: Lanati Kline DATE:
PROGRAM ID: CONC-DICTIONARY PAGE 1 OF 3

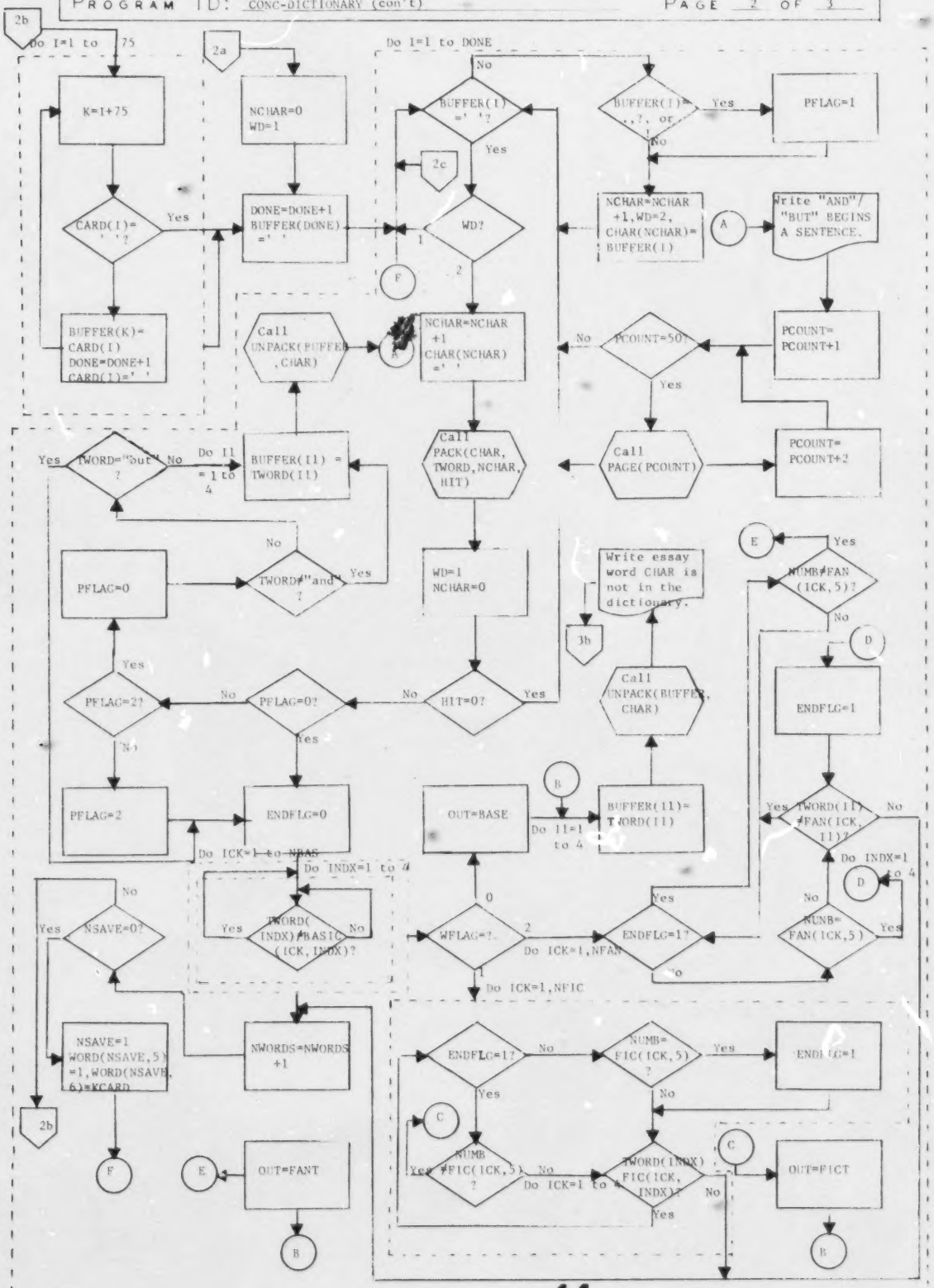


8.0-PROGRAM FLOWCHART

SWRL

FLOWCHART LAYOUT FORM

PROGRAMMER: Lanail Kline
PROGRAM ID: CONC-DICTIONARY (con't)
DATE:
PAGE 2 OF 3

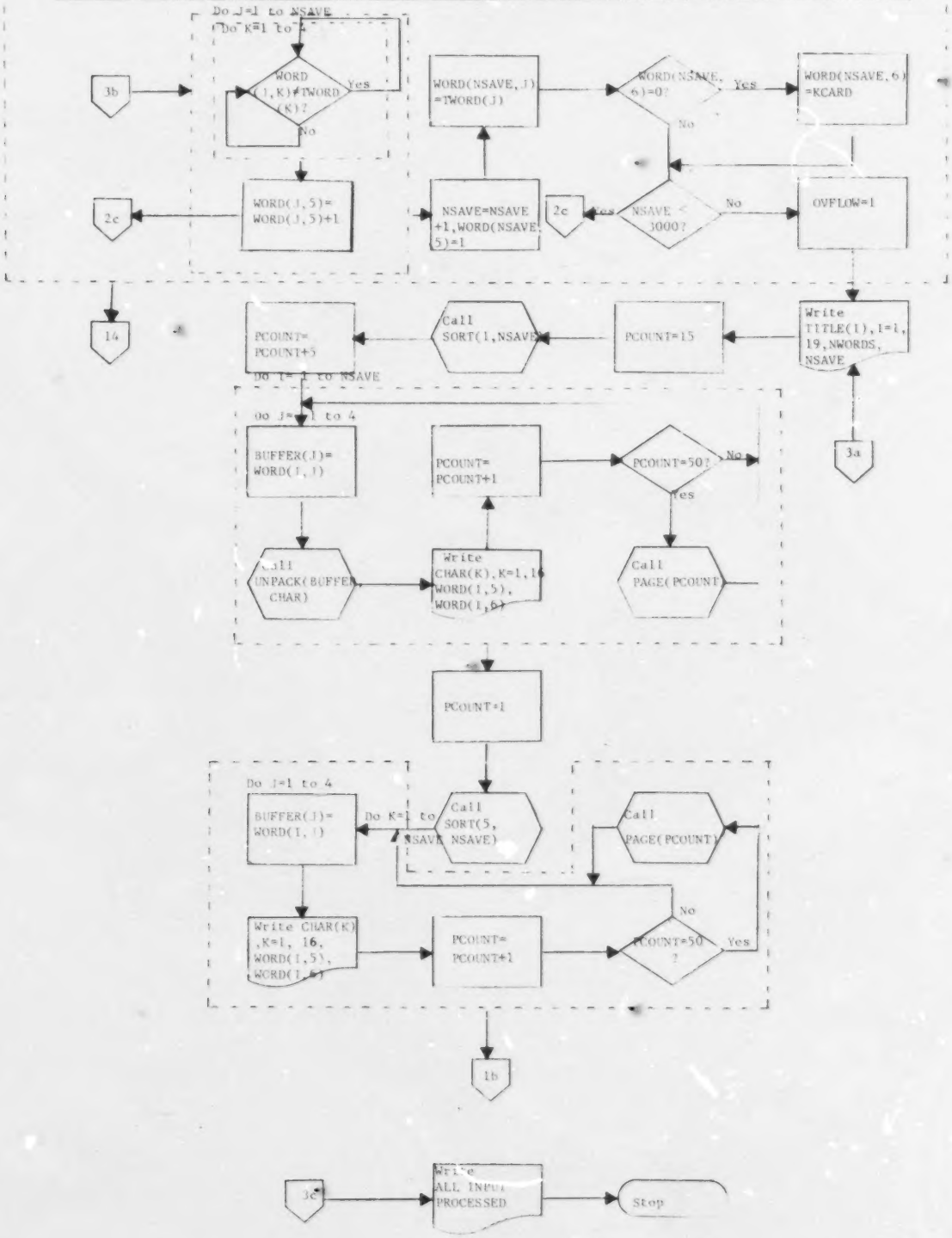


8.0-PROGRAM FLOWCHART

SWRL

FLOWCHART LAYOUT FORM

PROGRAMMER: Lanail Kline
PROGRAM ID: CONC-DICTIONARY (con't)
DATE:
PAGE 3 OF 3



```
C*****************************************************************
C         DAVID M. SHOEMAKER
C         SOUTHWEST REGIONAL LABORATORY FOR EDUCATIONAL RESEARCH
C           AND DEVELOPMENT
C         11300 LA CIENEGA BLVD
C         INGLEWOOD, CALIFORNIA
C
C         PROGRAM TO ALPHABETIZE AND LIST ACCORDING TO FREQUENCY OF
C         OCCURRENCE A SERIES OF WORDS ORGANIZED INTO A CONTENT BLOCK
C
C         ORGANIZATION OF DATA DECK
C
C         TITLE CARD (K CARDS)  CONTINUATION BY X IN COL 80
C         CONTENT DATA BLOCK (K CARDS)
C         DELIMITER CARD WITH * IN COL 1 (1 CARD)
C
C         PROGRAM WILL PROCESS K CONTENT BLOCKS
C*****************************************************************
      IMPLICIT INTEGER (A-Z)
      COMMON WORD(3000,6)
      DIMENSION NO(2)
      DIMENSION CARD(80),TITLE(38),BUFFER(200),TWORD(4),CHAR(50)
      DIMENSION TYPE(2),LWORD(2,16),BASIC(800,4),FAN(800,5),FIC(1500,5),
     +TEST1(4),TEST2(4),AND(4),BUT(4)
      DATA BLK,STAR/1H ,1H*/
      DATA          ENDLST/4HEND /,FICT/4HFIC /,FANT/4HFAN /,
     +BASE/4HBAS /,TEST1/1HA,1HN,1HD,1H /,TEST2/1HB,1HU,1HT,1H /
      DATA PER/1H./,QUES/1H?/,EXCLAM/1H!/
      WRITE(6,444)
444   FORMAT('1BEGIN PROCESSING')
C
C             INPUT THE WORD LISTS
C
      NBAS=0
      NB=0
      NFAN=0
      NFIC=0
10010 READ(5,10000) TYPE(1),NO(1),(LWORD(1,I1),I1=1,16),TYPE(2),NO(2),
     +(LWORD(2,I1),I1=1,16)
10000 FORMAT(A3,2X,A3,1X,16A1,14X,A3,2X,A3,1X,16A1)
      N1=1
10011 IF(TYPE(1).EQ.ENDLST) GO TO 10018
      IF(TYPE(N1).NE.BASE.AND.TYPE(N1).NE.FICT.AND.TYPE(N1).NE.FANT)
     +GO TO 10012
      NB=0
      DO 10007 I1=1,16
      IF(LWORD(N1,I1).EQ.BLK) GO TO 10008
      NB=NB+1
10007 CHAR(NB)=LWORD(N1,I1)
10008 NB=NB+1
      CHAR(NB)=BLK
      CALL PACK(CHAR,TWORD,NB,HIT)
      IF(TYPE(N1).EQ.FICT) GO TO 10003
      IF(TYPE(N1).EQ.FANT) GO TO 10004
C
C             BASIC WORD
C
10002 NBAS=NBAS+1
```

```
C
C               CHECK IF THE NUMBER OF BASIC INPUT WORDS HAVE EXCEEDED THE
C               DIMENSION
C
      IF(NBAS.GT.800) GO TO 8001
      DO 10009 I1=1,4
10009 BASIC(NBAS,I1)=TWORD(I1)
10015 IF(N1.EQ.2) GO TO 10010
      N1=2
      GO TO 10011
C
C               THE ARRAY BASIC, FIC, OR FAN WAS EXCEEDED
C               THE PROGRAM IS TERMINATED
C
8001  WRITE(6,8000) BASE
8000  FORMAT(' EXCEEDED ',A4,' ARRAY')
      GO TO 10006
8002  WRITE(6,8000) FICT
      GO TO 10006
8004  WRITE(6,8000) FANT
      GO TO 10006
10012 WRITE(6,10005) TYPE(1),NO(1),(LWORD(1,I1),I1=1,16),TYPE(2),NO(2),
     +(LWORD(2,I1),I1=1,16)
10005 FORMAT(' PROGRAM TEMINATED DUE TO ERROR ON TYPE FIELD IN THE WORD
     +LIST INPUT'/' ON CARD',1X,A3,2X,A3,1X,16A1,14X,A3,2X,A3,1X,16A1)
      GO TO 10006
C
C               FICTION WORD
C
10003 NFIC=NFIC+1
C
C               CHECK IF THE # OF FICTION INPUT WORDS HAVE EXCEEDED THE
C               DIMENSION
C
      IF(NFIC.GT.1500) GO TO 8002
      DO 10014 I1=1,4
10014 FIC(NFIC,I1)=TWORD(I1)
      FIC(NFIC,5)=NO(N1)
      GO TO 10015
C
C               FANTASY WORD
C
10004 NFAN=NFAN+1
C
C               CHECK IF THE # OF FANTASY INPUT WORDS HAVE EXCEEDED THE
C               DIMENSION
C
      IF(NFAN.GT.800) GO TO 8004
      DO 10017 I1=1,4
10017 FAN(NFAN,I1)=TWORD(I1)
      FAN(NFAN,5)=NO(N1)
      GO TO 10015
C
C               STORE THE WORDS 'AND' AND 'BUT'
10018 DO 470 I1=1,4
470   CHAR(I1)=TEST1(I1)
      CALL PACK(CHAR,TWORD,4,HIT)
      DO 10019 I1=1,4
```

```
10019 AND(I1)=TWORD(I1)
      DO 471 I1=1,4
471   CHAR(I1)=TEST2(I1)
      CALL PACK(CHAR,TWORD,4,HIT)
      DO 10020 I1=1,4
10020 BUT(I1)=TWORD(I1)
      PFLAG=0
      OVFLOW=0
 5000 IF ( OVFLOW .EQ. 1 ) GO TO 40
      WFLAG=1
      READ(5,4000,END=6000) THING,THING1,NUMB
4000  FORMAT(A3,A4,8X,A3)
      PCOUNT=0
      WRITE(6,7002) THING,THING1,NUMB
7002  FORMAT('1',A3,A4,' STORY NUMBER',A3)
      PCOUNT=PCOUNT+1
      IF(THING.EQ.FANT) GO TO 7000
      IF(THING.EQ.BASE) WFLAG=0
C
C             READ AND PRINT TITLE CARD
C
7001  READ(5,12) (TITLE(I),I=1,19),MORE
      WRITE(6,13) (TITLE(I),I=1,19)
      PCOUNT=PCOUNT+2
 20   IF ( MORE .EQ. BLK ) GO TO 25
      READ(5,12) (TITLE(I),I=20,38),MORE
      WRITE(6,13) (TITLE(I),I=20,38)
      PCOUNT=PCOUNT+2
      GO TO 20
C
C             ESSAY IS A FANTASY STORY
C
7000  WFLAG=2
      GO TO 7001
 25   WRITE (6,15)
      PCOUNT=PCOUNT+4
      KCARD=0
 40   OVFLOW=0
      DO 50 I=1,3000
      DO 50 J=1,6
 50   WORD(I,J)=0
      NSAVE=0
      NWORDS=0
      RDCARD=0
 1000 IF ( RDCARD .NE. 1 ) GO TO 90
      KCARD=KCARD+1
      GO TO 95
C
C             READ AND PRINT AN ESSAY CARD
C
90    READ(5,1) (CARD(I),I=1,75)
      KCARD=KCARD+1
      WRITE(6,9) (CARD(I),I=1,75)
      PCOUNT=PCOUNT+1
      IF(PCOUNT.EQ.50) CALL PAGE(PCOUNT)
      IF ( CARD(1) .EQ. STAR ) GO TO 250
C
C             BEGIN TO LOOK FOR WORDS ON ESSAY CARD
```

```
C
95    DONE=75
C
C             BUFFER CONTAINS THE INPUT CARD IMAGE
C
      RDCARD=0
      DO 100 I=1,75
 100  BUFFER(I)=CARD(I)
C
C             CHECKS IF WORD CONTINUES ONTO NEXT CARD
C
      IF(CARD(75).EQ.BLK) GO TO 130
C
C             READ AND PRINT AN ESSAY CARD
C
      READ(5,1) (CARD(I),I=1,75)
      KCARD=KCARD+1
      WRITE(6,9) (CARD(I),I=1,75)
      PCOUNT=PCOUNT+1
      IF(PCOUNT.EQ.50) CALL PAGE(PCOUNT)
      IF ( CARD(1) .EQ. STAR ) GO TO 250
      RDCARD=1
      DONE=75
      DO 120 I=1,75
      K=I+75
      IF ( CARD(I) .EQ. BLK ) GO TO 125
C
C             BUFFER CONTAINS THE REST OF THE WORD WHICH OVERFLOWED ONTO
C             THE NEXT CARD
C
      BUFFER(K)=CARD(I)
      DONE=DONE+1
      CARD(I)=BLK
 120  CONTINUE
125   DONE=DONE+1
      BUFFER(DONE)=BLK
C
C             PACKING SECTION
C
 130  NCHAR=0
      WD=1
      DO 200 I=1,DONE
      IF ( BUFFER(I) .EQ. BLK ) GO TO (200,160), WD
      IF(BUFFER(I).EQ.PER.OR.BUFFER(I).EQ.QUES.OR.BUFFER(I).EQ.EXCLAM)
     +PFLAG=1
      NCHAR=NCHAR+1
      WD=2
      CHAR(NCHAR)=BUFFER(I)
      GO TO 200
 160  NCHAR=NCHAR+1
      CHAR(NCHAR)=BLK
      CALL PACK (CHAR,TWORD,NCHAR,HIT)
      WD=1
      NCHAR=0
      IF ( HIT .EQ. 0 ) GO TO 200
C
C             CHECK IF PFLAG IS SET
C
```

```
      IF(PFLAG.EQ.0) GO TO 473
      IF(PFLAG.EQ.2) GO TO 1478
      PFLAG=2
      GO TO 473
C
C             CHECK IF THE WORD FOLLOWING PUNCUATION IS THE WORD AND
C             OR BUT
C
1478  PFLAG=0
      DO 474 I1=1,4
      IF(AND(I1).NE.TWORD(I1)) GO TO 1475
474   CONTINUE
      GO TO 1476
1475  DO 1474 I1=1,4
      IF(BUT(I1).NE.TWORD(I1)) GO TO 473
1474  CONTINUE
C
C             WORD EQUALS AND OR BUT
C
1476  DO 478 I1=1,4
478   BUFFER(I1)=TWORD(I1)
      CALL UNPACK(BUFFER,CHAR)
      WRITE(6,477) (CHAR(L),L=1,16)
477   FORMAT(' ',16A1,1X,'BEGINS A SENTENCE')
      PCOUNT=PCOUNT+1
      IF(PCOUNT.EQ.50) CALL PAGE(PCOUNT)
      GO TO 200
473   ENDFLG=0
      DO 400 ICK=1,NBAS
      DO 401 INDX=1,4
      IF(BASIC(ICK,INDX).NE.TWORD(INDX)) GO TO 400
401   CONTINUE
C
C             FOUND WORD IN BASIC WORD LIST
C
      GO TO 402
400   CONTINUE
C
C             WORD NOT IN BASIC WORD LIST
C
      IF(WFLAG.EQ.0) GO TO 7077
      IF(WFLAG.EQ.2) GO TO 403
C
C             CHECK IF WORD IS IN THE FICTION WORD LIST
C
      DO 404 ICK=1,NFIC
      IF(ENDFLG.EQ.1) GO TO 405
      IF(NUMB.EQ.FIC(ICK,5)) GO TO 406
C
C             PROCESS LIST
C
420   DO 407 INDX=1,4
      IF(FIC(ICK,INDX).NE.TWORD(INDX)) GO TO 404
407   CONTINUE
C
C             FOUND WORD IN FICTION WORD LIST
C
      GO TO 402
```

```
406   ENDFLG=1
      GO TO 420
C
C           CHECK IF AT END OF ALLOWABLE WORDS IN BASIC WORD LIST
C
405   IF(NUMB.NE.FIC(ICK,5)) GO TO 427
      GO TO 420
404   CONTINUE
427   OUT=FICT
409   DO 472 I1=1,4
472   BUFFER(I1)=TWORD(I1)
      CALL UNPACK(BUFFER,CHAR)
      WRITE(6,408) OUT,(CHAR(L),L=1,16)
408   FORMAT('0THIS WORD DOES NOT APPEAR IN THE BASIC WORD LIST OR IN TH
     +E ',A3,' WORD LIST',10X,16A1)
      PCOUNT=PCOUNT+2
      IF(PCOUNT.EQ.50) CALL PAGE(PCOUNT)
      GO TO 200
7077  OUT=BASE
      GO TO 409
C
C           CHECK IF WORD IS IN THE FANTASY WORD LIST
C
403   DO 410 ICK=1,NFAN
      IF(ENDFLG.EQ.1) GO TO 411
      IF(NUMB.EQ.FAN(ICK,5)) GO TO 412
C
C           PROCESS LIST
C
421   DO 413 INDX=1,4
      IF(FAN(ICK,INDX).NE.TWORD(INDX)) GO TO 410
413   CONTINUE
C
C           FOUND WORD IN FANTASY WORD LIST
C
      GO TO 402
412   ENDFLG=1
      GO TO 421
C
C           CHECK IF AT END OF ALLOWABLE WORDS IN FANTASY WORD LIST
C
411   IF(NUMB.NE.FAN(ICK,5)) GO TO 423
      GO TO 421
410   CONTINUE
423   OUT=FANT
      GO TO 409
480   NSAVE=1
      DO 481 J=1,4
481   WORD(NSAVE,J)=TWORD(J)
      WORD(NSAVE,5)=1
      WORD(NSAVE,6)=KCARD
      GO TO 200
402   NWORDS=NWORDS+1
      IF(NSAVE.EQ.0) GO TO 480
      DO 170 J=1,NSAVE
      DO 175 K=1,4
      IF ( WORD(J,K) .NE. TWORD(K) ) GO TO 170
175   CONTINUE
```

```
      WORD(J,5)=WORD(J,5)+1
      GO TO 200
  170 CONTINUE
      NSAVE=NSAVE+1
      DO 180 J=1,4
  180 WORD(NSAVE,J)=TWORD(J)
      WORD(NSAVE,5)=1
      IF ( WORD(NSAVE,6) .EQ. 0 ) WORD(NSAVE,6)=KCARD
      IF ( NSAVE .LT. 3000 ) GO TO 200
      WRITE (6,11)
      OVFLOW=1
      GO TO 250
  200 CONTINUE
      GO TO 1000
  250 CONTINUE
      WRITE(6,2) (TITLE(I),I=1,19),NWORDS,NSAVE
      PCOUNT=15
      CALL SORT (1,NSAVE)
      WRITE (6,3)
      PCOUNT=PCOUNT+5
      DO 300 I=1,NSAVE
      DO 310 J=1,4
  310 BUFFER(J)=WORD(I,J)
      CALL UNPACK (BUFFER,CHAR)
      WRITE (6,4) (CHAR(K),K=1,16),WORD(I,5),WORD(I,6)
      PCOUNT=PCOUNT+1
      IF(PCOUNT.EQ.50) CALL PAGE(PCOUNT)
  300 CONTINUE
      WRITE (6,5)
      PCOUNT=1
      CALL SORT (5,NSAVE)
      DO 350 I=1,NSAVE
      DO 360 J=1,4
  360 BUFFER(J)=WORD(I,J)
      CALL UNPACK (BUFFER,CHAR)
      WRITE (6,4) (CHAR(K),K=1,16),WORD(I,5),WORD(I,6)
      PCOUNT=PCOUNT+1
      IF(PCOUNT.EQ.50) CALL PAGE(PCOUNT)
  350 CONTINUE
      GO TO 5000
 6000 WRITE (6,10)
    1 FORMAT(75A1)
    2 FORMAT('1',19A4,A2////9X,'TOTAL WORDS   ',I8////9X,'UNIQUE WORDS
     1 ',I7////)
    3 FORMAT (24H ALPHABETICAL DICTIONARY//37X,4HFREQ,6X,4HCARD//)
    4 FORMAT (15X,16A1,2I10)
    5 FORMAT (19H1FREQUENCY PER WORD//37X,4HFREQ,6X,4HCARD//)
    9 FORMAT(15X,75A1,I10)
   10 FORMAT(20H1ALL INPUT PROCESSED)
   11 FORMAT (31H1MAX WORD TABLE LENGTH EXCEEDED////33H INTERMEDIATE RES
     1ULTS WILL FOLLOW////)
   12 FORMAT(18A4,A2,A1)
   13 FORMAT('0',18A4,A2)
   14 FORMAT (1X,19A4,A3)
   15 FORMAT (////)
10006 STOP
      END
```

1.0 - PROGRAM IDENTIFICATION

SUBROUTINE PACK

2.0 - OBJECTIVE

To encode a word of one to sixteen characters into four integers in order to conserve core and make sorting easier.

3.0 - PROGRAM DESCRIPTION

3.1 - Program Logic

The subroutine receives an array of up to sixteen characters and converts each character into an integer equivalent. The program then takes the numbers, four at a time, and calculates a unique integer for that sequence of numbers. Each integer is stored as an element of a four element array. The program uses the following algorithm to encode the word:

WORD(NWORD)=WORD(NWORD)+CHAR(I)*50**(5-NCHAR), where $1 \leq$ NWORD ≤ 4, $1 \leq$ NCHAR ≤ 4, and $1 \leq$ I ≤ 16.

3.2 - Variables and Internal Tables

Integers:

HIT	- is a flag which indicates that no non-blank characters were found in CHAR.
NC	- is an input argument which indicates the number of characters are in CHAR.
NWORD	- is an index to the array WORD. $1 \leq$ NWORD ≤ 4
NCHAR	- is a counter used in the packing algorithm. $1 \leq$ NCHAR ≤ 4
SWITCH	- is a flag which indicates that the last non-blank character in CHAR was packed and to return the program flow to the main program.
I	- is an index for CHAR. $1 \leq$ CHAR ≤ 16

Integer Arrays:

CHAR(50)	- is an input argument containing the unpacked word. There can be at most sixteen non-blank letters in CHAR.
WORD(4)	- is an output argument containing the packed word.
ALIST(29)	- is an array which contains the characters which can make up a word. The index of an element is the integer value assigned to a letter prior to packing, e.g., ALIST(2)='A' and CHAR(I)='A', then CHAR(I) will equal two just prior to packing.

3.3 - Input/Output Arguments

Input:

CHAR
NC } See Section 3.2

Output:

WORD
HIT } See Section 3.2

4.0 - SUBROUTINES AND FUNCTIONS

None

5.0 - DATA SPECIFICATIONS

Not Applicable

6.0 - PROGRAM CONSTRAINTS AND LIMITATIONS

6.1 - Programming Language
FORTRAN IV

6.2 - Vendor
University of California, Los Angeles
Campus Computing Network

6.3 - Storage Requirements
1018 Bytes
See Section 6.3 of the main program documentation.

6.4 - Hardware Configuration
IBM 360 Model 91

6.5 - Program Parameters
None

6.6 - Error Messages
None

7.0 - OPERATING INSTRUCTIONS

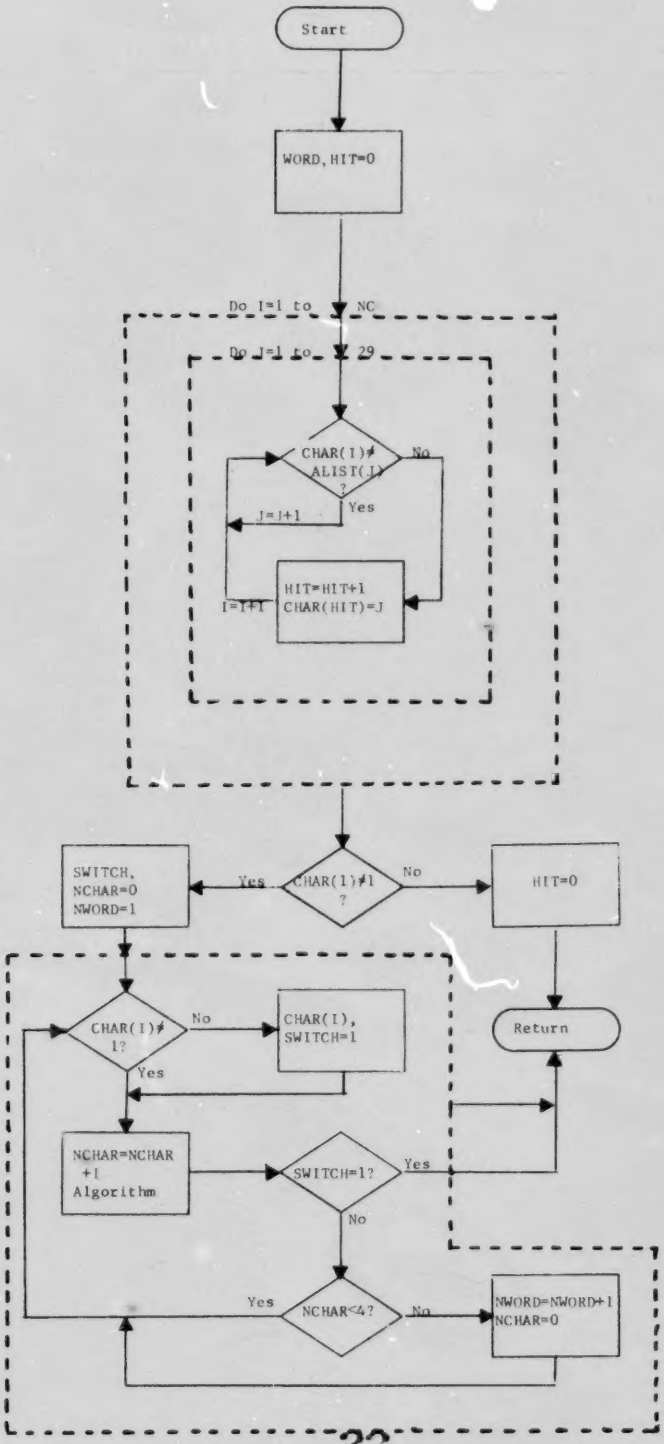
See Section 7.0 of the main program documentation.

8.0-PROGRAM FLOWCHART

SWRL

FLOWCHART LAYOUT FORM

PROGRAMMER: Lanaii Kline DATE:
PROGRAM ID: SUBROUTINE PACK PAGE 1 OF 1



```
0001          SUBROUTINE PACK (CHAR,WORD,NC,HIT)
0002          IMPLICIT INTEGER (A-Z)
0003          DIMENSION CHAR(50),WORD(4),ALIST(29)
0004          DATA ALIST/1H ,1HA,1HB,1HC,1HD,1HE,1HF,1HG,1HH,1HI,1HJ,1HK,1HL,
             11HM,1HN,1HO,1HP,1HQ,1HR,1HS,1HT,1HU,1HV,1HW,1HX,1HY,1HZ,1H',1H-/
0005          DO 50 I=1,4
0006       50 WORD(I)=0
0007          HIT=0
0008          DO 100 I=1,NC
0009          DO 110 J=1,29
0010          IF ( CHAR(I) .NE. ALIST(J) ) GO TO 110
0011          HIT=HIT+1
0012          CHAR(HIT)=J
0013          GO TO 100
0014      110 CONTINUE
0015      100 CONTINUE
0016          IF ( CHAR(1) .NE. 1 ) GO TO 115
0017          HIT=0
0018          RETURN
0019      115 SWITCH=0
0020          NWORD=1
0021          NCHAR=0
0022          DO 200 I=1,16
0023          IF ( CHAR(I) .NE. 1 ) GO TO 210
0024          CHAR(I)=1
0025          SWITCH=1
0026      210 NCHAR=NCHAR+1
0027          WORD(NWORD)=WORD(NWORD)+CHAR(I)*50 **(5-NCHAR)
0028          IF ( SWITCH .EQ. 1 ) RETURN
0029          IF ( NCHAR .LT. 4 ) GO TO 200
0030          NWORD=NWORD+1
0031          NCHAR=0
0032      200 CONTINUE
0033          RETURN
0034          END
```

1.0 - PROGRAM IDENTIFICATION

SUBROUTINE UNPACK

2.0 - OBJECTIVE

To decode the packed words into characters for printing.

3.0 - PROGRAM DESCRIPTION

3.1 - Program Logic

The program unpacks the four integer elements in WORD into one to sixteen character words and stores the characters in CHAR. The numbers are unpacked using the following algorithm:

CHAR(NCHAR)=(BASE/(50**(5-J)))-(BASE/(50**(6-J)))*50,
where $1 \leq$ NCHAR ≤ 16; BASE=WORD(I), $(1 \leq I \leq 4)$; and $1 \leq J \leq 4$.

3.2 - Variables and Internal Tables

Integers:

START	- is a counter which indicates when the word is unpacked and the rest of the word is blank. START=NCHAR+1 and $1 \leq$ START ≤ 16
NCHAR	- is a counter which indicates where the last non-blank character is in WORD.
BASE	- is the current value of WORD(I).
J	- is an index used in unpacking a word. $1 \leq J \leq 4$
NN	- is a counter used in unpacking a word. $1 \leq NN \leq 4$

Integer Arrays:

WORD(200)	- is the input argument which contains the integer values of the word to be decoded.
CHAR(50)	- is the output argument which will contain the characters of the decoded word.
ALIST(29)	- is an array containing the characters which can make up a legal word.

3.3 - Input/Output Arguments

Input:

WORD

Output:

CHAR

See Section 3.2

4.0 - SUBROUTINES AND FUNCTIONS

None

5.0 - DATA SPECIFICATIONS

Not Applicable

6.0 - PROGRAM CONSTRAINTS AND LIMITATIONS

6.1 - Programming Language
FORTRAN IV

6.2 - Vendor
University of California, Los Angeles
Campus Computing Network

6.3 - Storage Requirements
1058 Bytes
See Section 6.3 of the main program documentation.

6.4 - Hardware Configuration
IBM 360 Model 91

6.5 - Program Parameters
None

6.6 - Error Messages
None

7.0 - OPERATING INSTRUCTIONS

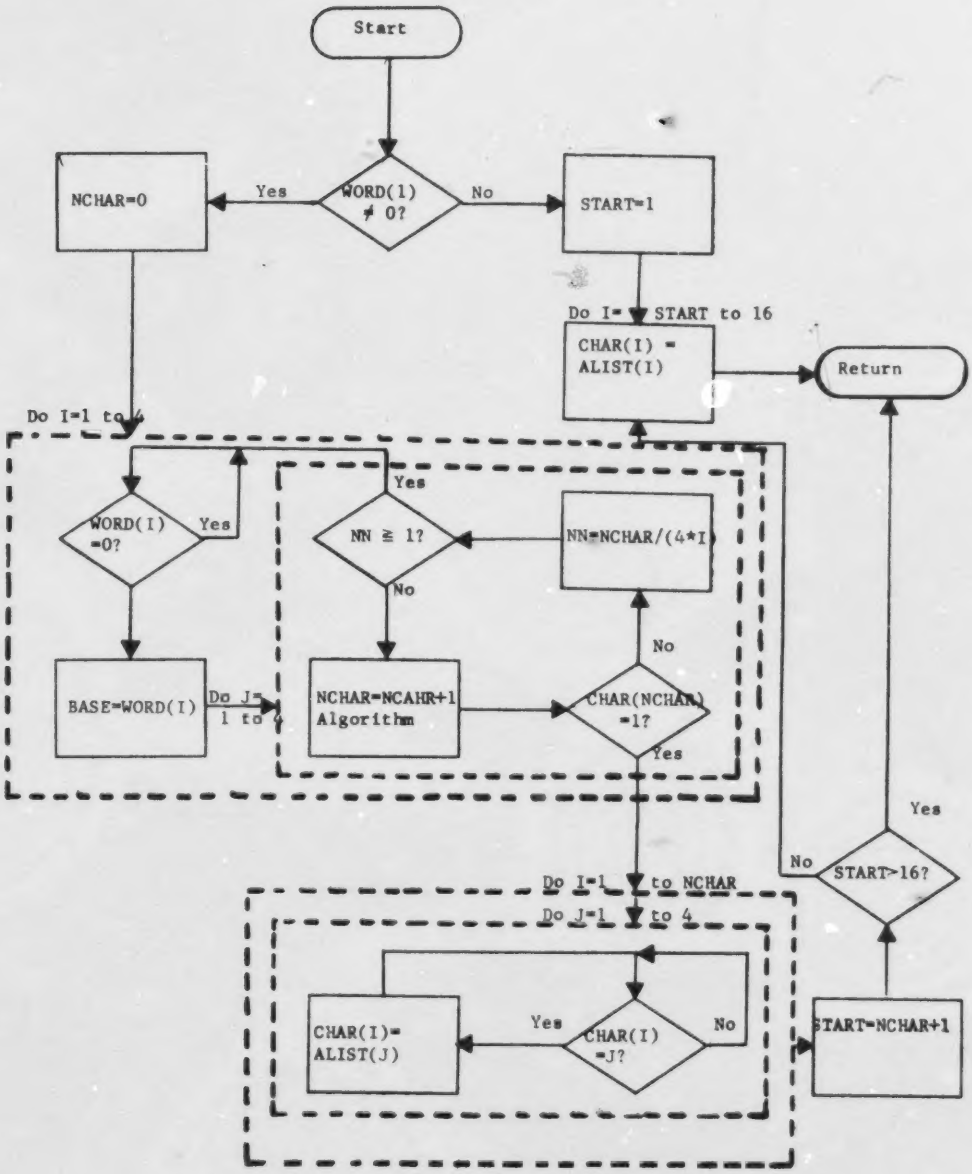
See Section 7.0 of the main program documentation.

8.0-PROGRAM FLOWCHART

SWRI

FLOWCHART LAYOUT FORM

PROGRAMMER: Lanaii Kline
PROGRAM ID: SUBROUTINE UNPACK
DATE:
PAGE 1 OF 1



FORTRAN IV C LEVEL 20 UNPACK DATE = 71363 22/42/38

```
0001          SUBROUTINE UNPACK (WORD,CHAR)
0002          IMPLICIT INTEGER (A-Z)
0003          DIMENSION WORD(200),CHAR(50),ALIST(29)
0004          DATA ALIST/1H ,1HA,1HB,1HC,1HD,1HE,1HF,1HG,1HH,1HI,1HJ,1HK,1HL,
             11HM,1HN,1HO,1HP,1HQ,1HR,1HS,1HT,1HU,1HV,1HW,1HX,1HY,1HZ,1H',1H-/
0005          IF ( WORD(1) .NE. 0 ) GO TO 50
0006          START=1
0007          GO TO 350
0008     50   NCHAR=0
0009          DO 100 I=1,4
0010          IF ( WORD(I) .EQ. 0 ) GO TO 100
0011          BASE=WORD(I)
0012          DO 150 J=1,4
0013          NCHAR=NCHAR+1
0014          CHAR(NCHAR)=(BASE/(50 **(5-J)))-(BASE/(50 **(6-J)))*50
0015          IF ( CHAR(NCHAR) .EQ.  1 ) GO TO 200
0016          NN=NCHAR/(4*I)
0017          IF ( NN .GE. 1 ) GO TO 100
0018    150   CONTINUE
0019    100   CONTINUE
0020    200   DO 400 I=1,NCHAR
0021          DO 300 J=1,29
0022          IF ( CHAR(I) .EQ. J ) CHAR(I)=ALIST(J)
0023    300   CONTINUE
0024          START=NCHAR+1
0025          IF ( START .GE. 16 ) RETURN
0026    350   DO 400 I=START,16
0027    400   CHAR(I)=ALIST(1)
0028          RETURN
0029          END
```

1.0 - PROGRAM IDENTIFICATION

SUBROUTINE SORT

2.0 - OBJECTIVE

To sort the unique word list alphabetically if COL equals one, or by frequency count if COL equals five.

3.0 - PROGRAM DESCRIPTION

3.1 - Program Logic
If the input argument COL equals one, the subroutine sorts alphabetically on the first four characters of the words in the array WORD. If COL equals five, the subroutine sorts the array WORD numerically on the frequency counts.

3.2 - Variables and Internal Tables
Integers:

DONE - is a variable whose value is one less than NROWS.
NROWS - is the number of words to be sorted.
COL - is a flag which indicates which column on which to sort.
TEMP - is a dummy variable which is used to exchange the variables which are out of order.

Integer Arrays:

WORD(3000,6) - is the array containing the unique words and its frequency counts.

4.0 - SUBROUTINES AND FUNCTIONS

None

5.0 - DATA SPECIFICATIONS

Not Applicable

6.0 - PROGRAM CONSTRAINTS AND LIMITATIONS

6.1 - Programming Language
FORTRAN IV

6.2 - Vendor
University of California, Los Angeles
Campus Computing Network

6.3 - Storage Requirements
680 Bytes
See Section 6.3 of the main program documentation.

6.4 - Hardware Configuration
IBM 360 Model 91

6.5 - Program Parameters
None

6.6 - Error Messages
None

7.0 - OPERATING INSTRUCTIONS

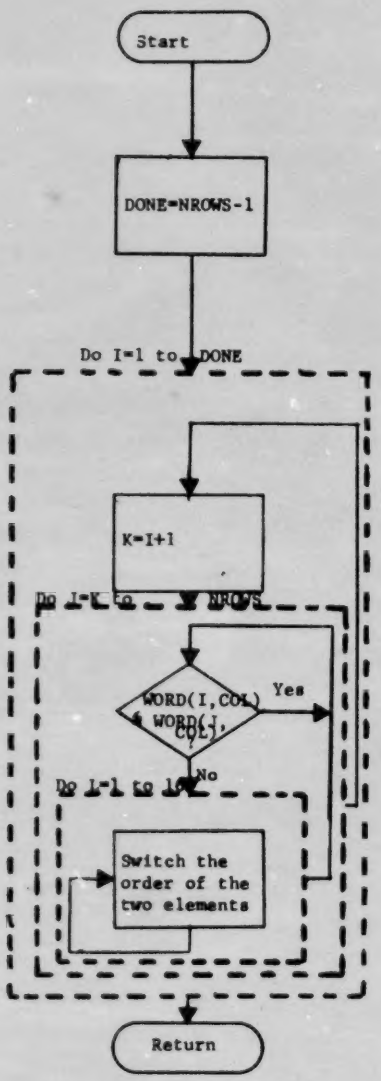
See Section 7.0 of main program documentation.

8.0-PROGRAM FLOWCHART

SWRI

FLOWCHART LAYOUT FORM

PROGRAMMER: Lanaii Kline
PROGRAM ID: SUBROUTINE SORT
DATE:
PAGE 1 OF 1



```
FORTRAN IV C LEVEL  20                SCRT                DATE = 71363          22/42/38

                  SUBROUTINE SCRT (COL,NROWS)
                  IMPLICIT INTEGER (A-Z)
                  COMMON WORD(3000,6)
                  DONE=NROWS-1
                  DO 100 I=1,DONE
                  K=I+1
                  DO 100 J=K,NROWS
                  IF ( WORD(I,COL) .LE. WORD(J,COL) ) GO TO 100
                  DO 120 L=1,6
                  TEMP=WORD(J,L)
                  WORD(J,L)=WORD(I,L)
            120   WORD(I,L)=TEMP
            100   CONTINUE
                  RETURN
                  END
```

1.0 - PROGRAM IDENTIFICATION

SUBROUTINE PAGE

2.0 - OBJECTIVE

To cause a top of form on the line printer. This is used when using the FORTRAN G Compiler at UCLA CCN, although it can be used with any other compiler.

3.0 - PROGRAM DESCRIPTION

3.1 - Program Logic
The program writes a top of form and reinitializes the line count to zero.

3.2 - Variables and Internal Tables
Integers:

PCOUNT - is a variable which counts the number of lines printed. PCOUNT equals zero when the subroutine returns control to the main program.

3.3 - Input/Output Arguments
Input and Output:

PCOUNT - See Section 3.2

4.0 - SUBROUTINES AND FUNCTIONS

None

5.0 - DATA SPECIFICATIONS

5.1 - Input Formats
None

5.2 - Output Formats
('1') - page eject

6.0 - PROGRAM CONSTRAINTS

6.1 - Programming Language
FORTRAN IV

6.2 - Vendor
University of California, Los Angeles
Campus Computing Network

6.3 - Storage Requirements
312 Bytes
See Section 6.3 of the main program documentation.

6.4 - Hardware Configuration
IBM 360 Model 91, Line Printer

6.5 - Program Parameter
None

6.6 - Error Messages
None

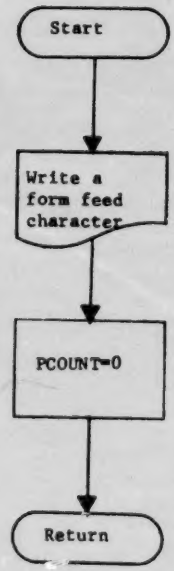
7.0 - OPERATING INSTRUCTIONS

See Section 7.0 of the main program documentation.

SWRI

FLOWCHART LAYOUT FORM

PROGRAMMER: Lanail Kline
PROGRAM ID: SUBROUTINE PAGE
DATE:
PAGE 1 OF 1



SWRI

COMPUTER SYSTEMS
FILE/RECORD LAYOUT

TITLE: CONC-DICTIONARY DATE:

FILE ID: Input Formats

PROGRAMMER: Lanaii Kline

RECORDING MODE:
FIXED [x] LENGTH 80
VARIABLE [ ] MAX MIN

BLOCKING FACTOR:
BLOCK CONTAINS [3600] RECORDS

REMARKS:

CARD LAYOUT [X]
TAPE
RECORD LAYOUT [ ]

FORMAT LEGEND:

△ = Blank	P = Packed Dec.
A = Alpha	O = Octal
N = Numeric	H = Hexadecimal
X = Alpha/Num.	B = Binary

RECORD I.D. Dictionary

Field	Type		Number		Word		Type		Number		Word	
FORMAT	A	△	A	△	A	△	A	△	A	△	A	△

POSITIONS 5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95 100

☐ CONTINUED

RECORD I.D. Delimiter

Field	END	
FORMAT	A	△

POSITIONS 5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95 100

☐ CONTINUED

RECORD I.D. Indentification

Field	Type	Filler	Filler	Number	
FORMAT	A	A	△	A	△

POSITIONS 5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95 100

☐ CONTINUED

RECORD I.D. Title

Field	Title	Name	
FORMAT	A	△	△

SWRI

COMPUTER SYSTEMS
FILE/RECORD LAYOUT

TITLE: CONC-DICTIONARY DATE:

FILE ID: Input Formats (Cont.)

PROGRAMMER: Lanaii Kline

RECORDING MODE:
FIXED [X]
LENGTH 80
VARIABLE [ ]
MAX
MIN

BLOCKING FACTOR:
BLOCK CONTAINS
[3600] RECORDS

REMARKS:

CARD LAYOUT [X]
TAPE
RECORD LAYOUT [ ]

FORMAT LEGEND:

△ = Blank	P = Packed Dec.
A = Alpha	O = Octal
N = Numeric	H = Hexadecimal
X = Alpha/Num.	B = Binary

RECORD I.D.
Essay

Words

FORMAT A △

POSITIONS 5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95 100

[ ] CONTINUED

RECORD I.D.
Delimiter

FORMAT A △

POSITIONS 5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95 100

[ ] CONTINUED

RECORD I.D.

FORMAT

POSITIONS 5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95 100

[ ] CONTINUED

RECORD I.D.

FORMAT

SWRI

COMPUTER SYSTEMS
FILE/RECORD LAYOUT

TITLE: CONC-DICTIONARY DATE:

FILE ID: Output Formats

PROGRAMMER: Lanaii Kline

RECORDING MODE:
FIXED [X]
LENGTH 132
VARIABLE [ ]
MAX
MIN

BLOCKING FACTOR:
BLOCK CONTAINS
[ ] RECORDS

REMARKS:

PRINTER
CARD LAYOUT [X]

TAPE
RECORD LAYOUT [ ]

FORMAT LEGEND:

△ = Blank	P = Packed Dec.
A = Alpha	O = Octal
N = Numeric	H = Hexadecimal
X = Alpha/Num.	B = Binary

RECORD I.D.
Identification

	Type	Filler	'STORY NUMBER'	Story Number
FORMAT	A	A	A	A
POSITIONS	5	10	15	20

■ CONTINUED

RECORD I.D.

FORMAT △

POSITIONS 5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95 100

□ CONTINUED

RECORD I.D.
Title (may be more than one record long)

TITLE

FORMAT A △

POSITIONS 5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95 100

■ CONTINUED

RECORD I.D.

FORMAT △

SWRI

COMPUTER SYSTEMS
FILE/RECORD LAYOUT

TITLE: CONC-DICTIONARY DATE:

FILE ID: Output Formats (Cont.)

PROGRAMMER: Lanaii Kline

RECORDING MODE:
FIXED [X]
LENGTH 132
VARIABLE [ ]
MAX
MIN

BLOCKING FACTOR:
BLOCK CONTAINS
RECORDS

REMARKS:

PRINTER
CARD LAYOUT [X]

TAPE
RECORD LAYOUT [ ]

FORMAT LEGEND:

△ = Blank	P = Packed Dec.
A = Alpha	O = Octal
N = Numeric	H = Hexadecimal
X = Alpha/Num.	B = Binary

RECORD I.D. Essay (Repeat this record format for each essay line printed.)		Essay Line	Card Number
FORMAT	△	A	N
POSITIONS	5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95 100		

■ CONTINUED

RECORD I.D.		
FORMAT	△	
POSITIONS	5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95 100	

□ CONTINUED

RECORD I.D. Illegal Word Message (Repeat this record format for each illegal word encountered in	'THIS WORD DOES NOT APPEAR IN THE BASIC WORD LIST OR IN THE '	Dictionary	' WORD LIST'		Word
FORMAT	A	A	A	△	A	△
POSITIONS	5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95 100				

■ CONTINUED

RECORD I.D. the above essay record.)		
FORMAT	△	

SWRI

COMPUTER SYSTEMS
FILE/RECORD LAYOUT

TITLE: CONC-DICTIONARY DATE:

FILE ID: Output Formats (Cont.)

PROGRAMMER: Lanaii Kline

RECORDING MODE:
FIXED [X] LENGTH 132
VARIABLE [ ]
MAX
MIN

BLOCKING FACTOR:
BLOCK CONTAINS [ ] RECORDS

REMARKS:

PRINTER CARD LAYOUT [X]
TAPE RECORD LAYOUT [ ]

FORMAT LEGEND:

△ = Blank	P = Packed Dec.
A = Alpha	O = Octal
N = Numeric	H = Hexadecimal
X = Alpha/Num.	B = Binary

RECORD I.D.
Title

TITLE

FORMAT: A △
POSITIONS: 5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95 100

■ CONTINUED
RECORD I.D.

3 blank lines

FORMAT: △
POSITIONS: 5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95 100

□ CONTINUED
RECORD I.D.
Total Words

	'TOTAL WORDS ¥¥¥'	Number of Words

FORMAT: △ A N △
POSITIONS: 5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95 100

■ CONTINUED
RECORD I.D.

3 blank lines

FORMAT: △

SWRI

COMPUTER SYSTEMS
FILE/RECORD LAYOUT

RECORDING MODE:
FIXED [X] LENGTH 132
VARIABLE [ ]
MAX
MIN

BLOCKING FACTOR:
BLOCK CONTAINS [ ] RECORDS

REMARKS:

PRINTER
CARD LAYOUT [X]

TAPE
RECORD LAYOUT [ ]

FORMAT LEGEND:
△ = Blank
A = Alpha
N = Numeric
X = Alpha/Num.
P = Packed Dec
O = Octal
H = Hexadecima
B = Binary

TITLE: CONC-DICTIONARY DATE:

FILE ID: Output Formats (Cont.)

PROGRAMMER: Lanaii Kline

RECORD I.D. Unique Words		'UNIQUE WORDS ¥¥¥'	Number of Unique Words
FORMAT	△	A	N
POSITIONS	5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95 10		

■ CONTINUED

RECORD I.D.		4 blank lines
FORMAT	△	
POSITIONS	5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95 10	

□ CONTINUED

RECORD I.D. Alphabetized word list Title record	'ALPHABETIZED DICTIONARY'	
FORMAT	A	△
POSITIONS	5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95 10	

■ CONTINUED

RECORD I.D.		1 blank line
FORMAT	△	

SWRI

COMPUTER SYSTEMS FILE/RECORD LAYOUT

TITLE: CONC-DICTIONARY DATE:

FILE ID: Output Formats (Cont.)

PROGRAMMER: Lanaii Kline

RECORDING MODE:
FIXED [X] LENGTH 132
VARIABLE [ ] MAX MIN
BLOCKING FACTOR:
BLOCK CONTAINS [ ] RECORDS

REMARKS:

PRINTER CARD LAYOUT [X]
TAPE RECORD LAYOUT [ ]

FORMAT LEGEND:

△ = Blank	P = Packed Dec.
A = Alpha	O = Octal
N = Numeric	H = Hexadecimal
X = Alpha/Num.	B = Binary

RECORD I.D. Headings for alphabetized word list

Positions	1–37	38–41	42–47	48–51	52–100
Content		'FREQ'		'CARD'	
FORMAT	△	A	△	A	△

■ CONTINUED

RECORD I.D.

Positions	1–31	32–100
Content		2 blank lines
FORMAT	△	

☐ CONTINUED

RECORD I.D. Word List

Positions	1–15	16–31	32–41	42–51	52–100
Content		Word	Frequency	Card on which the word first appeared	
FORMAT	△	A	N	N	△

■ CONTINUED

RECORD I.D.

FORMAT △

SWRL

COMPUTER SYSTEMS
FILE/RECORD LAYOUT

TITLE: CONC-DICTIONARY DATE:

FILE ID: Output Formats (Cont.)

PROGRAMMER: Lanaii Kline

RECORDING MODE:
FIXED [X]
LENGTH 132
VARIABLE [ ]
MAX
MIN

BLOCKING FACTOR:
BLOCK CONTAINS
[ ] RECORDS

REMARKS:

PRINTER
CARD LAYOUT [X]

TAPE
RECORD LAYOUT [ ]

FORMAT LEGEND:

△ = Blank	P = Packed Dec.
A = Alpha	O = Octal
N = Numeric	H = Hexadecimal
X = Alpha/Num.	B = Binary

RECORD I.D.
Heading for frequency
ordered word list

'FREQUENCY
PER WORD'

FORMAT A △
POSITIONS 5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95 100

■ CONTINUED
RECORD I.D.

2 blank lines

FORMAT △
POSITIONS 5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95 100

□ CONTINUED
RECORD I.D.
Heading for frequency
ordered word list

'FREQ'

'CARD'

FORMAT △ A △ A △
POSITIONS 5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95 100

■ CONTINUED
RECORD I.D.

FORMAT △

FORTRAN IV G LEVEL 20 PAGE DATE = 713

```
0001          SUBROUTINE PAGE(PCOUNT)
0002          INTEGER PCOUNT
0003          WRITE(6,1)
0004    1     FORMAT('1')
0005          PCOUNT=0
0006          RETURN
0007          END
```

BASIC	A
BASIC	AM
BASIC	AND
BASIC	ANN'S
BASIC	ANTS
BASIC	ARE
BASIC	ASKS
BASIC	ASKING
BASIC	BACK
BASIC	BACKED
BASIC	BAD
BASIC	BAMS
BASIC	BAMMING
BASIC	BANDS
BASIC	BANDED
BASIC	BANK
BASIC	BANK'S
BASIC	BANKING
BASIC	BEING
BASIC	BEDS
BASIC	BEDDED
BASIC	BELL
BASIC	BELL'S
BASIC	BELTS
BASIC	BELTED
BASIC	BEND
BASIC	BENDING
BASIC	BEST
BASIC	BOXES
BASIC	BOXED
BASIC	BUD
BASIC	BUT
BASIC	BUTTING
BASIC	CAMPS
BASIC	CAMPED
BASIC	CAN
BASIC	CAN'S
BASIC	CANNING
BASIC	CRACKS
BASIC	CRACKED
BASIC	CRASH
BASIC	CRASHED
BASIC	CUT
BASIC	CUTTING
BASIC	DASHES
BASIC	DASHED
BASIC	DEN
BASIC	DEN'S
BASIC	DIG
BASIC	DIGGING
BASIC	DIPS
BASIC	DIPPING
BASIC	DISHES
BASIC	DISHED
BASIC	DOT
BASIC	DOT'S
BASIC	DOTTING
BASIC	DRUM

BASIC	ALL
BASIC	AN
BASIC	ANN
BASIC	ANT
BASIC	ANT'S
BASIC	ASK
BASIC	ASKED
BASIC	AT
BASIC	BACKS
BASIC	BACKING
BASIC	BAM
BASIC	BAMMED
BASIC	BAND
BASIC	BAND'S
BASIC	BANDING
BASIC	BANKS
BASIC	BANKED
BASIC	BE
BASIC	BED
BASIC	BED'S
BASIC	BEDDING
BASIC	BELLS
BASIC	BELT
BASIC	BELT'S
BASIC	BELTING
BASIC	BENDS
BASIC	BENT
BASIC	BOX
BASIC	BOX'S
BASIC	BOXING
BASIC	BUD'S
BASIC	BUTTED
BASIC	CAMP
BASIC	CAMP'S
BASIC	CAMPING
BASIC	CANS
BASIC	CANNED
BASIC	CRACK
BASIC	CRACK'S
BASIC	CRACKING
BASIC	CRASHES
BASIC	CRASHING
BASIC	CUTS
BASIC	DASH
BASIC	DASH'S
BASIC	DASHING
BASIC	DENS
BASIC	DID
BASIC	DIGS
BASIC	DIP
BASIC	DIPPED
BASIC	DISH
BASIC	DISH'S
BASIC	DISHING
BASIC	DOTS
BASIC	DOTTED
BASIC	DOWN
BASIC	DRUMS

BASIC	DRUM'S	BASIC	DRUMMED
BASIC	DRUMMING	BASIC	DUCK
BASIC	DUCKS	BASIC	DUCK'S
BASIC	DUCKED	BASIC	DUCKING
BASIC	DUST	BASIC	DUSTS
BASIC	DUSTED	BASIC	DUSTING
BASIC	FELL	BASIC	FISH
BASIC	FISHES	BASIC	FISH'S
BASIC	FISHED	BASIC	FISHING
BASIC	FIX	BASIC	FIXED
BASIC	FIXES	BASIC	IS
BASIC	FIXING	BASIC	FROM
BASIC	FUN	BASIC	GET
BASIC	GETS	BASIC	GETTING
BASIC	GLAD	BASIC	GO
BASIC	GOES	BASIC	GOING
BASIC	GRASS	BASIC	HAD
BASIC	HAM	BASIC	HAMS
BASIC	HAS	BASIC	HE
BASIC	HELP	BASIC	HELPS
BASIC	HELPED	BASIC	HELPING
BASIC	HID	BASIC	HILL
BASIC	HILLS	BASIC	HIM
BASIC	HIS	BASIC	HOP
BASIC	HOPS	BASIC	HOPPED
BASIC	HOPPING	BASIC	HOT
BASIC	I	BASIC	IN
BASIC	IT	BASIC	ITS
BASIC	JOB	BASIC	JOBS
BASIC	JUMP	BASIC	JUMPS
BASIC	JUMPED	BASIC	JUMPING
BASIC	JUST	BASIC	KEEP
BASIC	KEEPS	BASIC	KEEPING
BASIC	LAND	BASIC	LANDS
BASIC	LAND'S	BASIC	LANDED
BASIC	LANDING	BASIC	LEND
BASIC	LENDS	BASIC	LENDING
BASIC	LET	BASIC	LETS
BASIC	LETTING	BASIC	LID
BASIC	LIDS	BASIC	LIDS
BASIC	LIL	BASIC	LIL'S
BASIC	LIT	BASIC	LOCK
BASIC	LOCKS	BASIC	LOCKED
BASIC	LOCKING	BASIC	LOCK'S
BASIC	LOG	BASIC	LOGS
BASIC	LOG'S	BASIC	LOGGED
BASIC	LOGGING	BASIC	LOT
BASIC	LOTS	BASIC	LUCK
BASIC	MAN	BASIC	MAN'S
BASIC	MAP	BASIC	MAPS
BASIC	MAPS	BASIC	MAPPED
BASIC	MAPPING	BASIC	MAT
BASIC	MATS	BASIC	MAT'S
BASIC	MATTED	BASIC	MATTING
BASIC	MASK	BASIC	MASKS
BASIC	MASK'S	BASIC	MASKED
BASIC	MASKING	BASIC	ME
BASIC	MELT	BASIC	MELTS

BASIC	MELTED	BASIC	MELTING
BASIC	MET	BASIC	MOP
BASIC	MOPS	BASIC	MOP'S
BASIC	MOPPED	BASIC	MOPPING
BASIC	MUD	BASIC	MUST
BASIC	NAP	BASIC	NAPS
BASIC	NAPPED	BASIC	NAPPING
BASIC	NEED	BASIC	NEEDS
BASIC	NEEDED	BASIC	NEEDING
BASIC	NEST	BASIC	NESTS
BASIC	NEST'S	BASIC	NESTED
BASIC	NESTING	BASIC	NET
BASIC	NETS	BASIC	NET'S
BASIC	NETTED	BASIC	NETTING
BASIC	NEXT	BASIC	NIP
BASIC	NIPS	BASIC	NIP'S
BASIC	NIPPED	BASIC	NIPPING
BASIC	NOT	BASIC	NAT
BASIC	NAT'S	BASIC	OF
BASIC	ON	BASIC	OUT
BASIC	OUTING	BASIC	PACK
BASIC	PACKS	BASIC	PACK'S
BASIC	PACKED	BASIC	PACKING
BASIC	PAM	BASIC	PAM'S
BASIC	PAN	BASIC	PANS
BASIC	PANNED	BASIC	PANNING
BASIC	PAT	BASIC	PATS
BASIC	PAT'S	BASIC	PATTED
BASIC	PATTING	BASIC	PATH
BASIC	PATHS	BASIC	PEN
BASIC	PENS	BASIC	PEN'S
BASIC	PENNED	BASIC	PENNING
BASIC	PICK	BASIC	PICKS
BASIC	PICKED	BASIC	PICKING
BASIC	PIN	BASIC	PINS
BASIC	PIN'S	BASIC	PINNED
BASIC	PINNING	BASIC	PIP
BASIC	PIP'S	BASIC	PIT
BASIC	PITS	BASIC	PIT'S
BASIC	PITTED	BASIC	PITTING
BASIC	PLAY	BASIC	PLAYS
BASIC	PLAYED	BASIC	PLAYING
BASIC	POND	BASIC	PONDS
BASIC	POND'S	BASIC	POP
BASIC	POPS	BASIC	POPPED
BASIC	POPPING	BASIC	PUT
BASIC	PUTS	BASIC	PUTTING
BASIC	RENT	BASIC	RENTS
BASIC	RENTED	BASIC	RENTING
BASIC	REST	BASIC	RESTS
BASIC	RESTED	BASIC	RESTING
BASIC	ROCK	BASIC	ROCKS
BASIC	ROCKED	BASIC	ROCKING
BASIC	RUFF	BASIC	RUFF'S
BASIC	RUFFED	BASIC	RUFFING
BASIC	RUN	BASIC	RUNS
BASIC	RUNNING	BASIC	SACK
BASIC	SACKS	BASIC	SACK'S

BASIC	SACKED	BASIC	SACKING
BASIC	SAD	BASIC	SAID
BASIC	SANK	BASIC	SAT
BASIC	SEE	BASIC	SEES
BASIC	SEEING	BASIC	SELL
BASIC	SELLS	BASIC	SELLING
BASIC	SEND	BASIC	SENDS
BASIC	SENDING	BASIC	SENT
BASIC	SET	BASIC	SETS
BASIC	SETTING	BASIC	SHE
BASIC	SHELL	BASIC	SHELLS
BASIC	SHELL'S	BASIC	SHELLED
BASIC	SHELLING	BASIC	SHIP
BASIC	SHIPS	BASIC	SHIP'S
BASIC	SHIPPED	BASIC	SHIPPING
BASIC	SHOP	BASIC	SHOPS
BASIC	SHOP'S	BASIC	SHOPPED
BASIC	SHOPPING	BASIC	SHUT
BASIC	SHUTS	BASIC	SHUTTING
BASIC	SICK	BASIC	SINK
BASIC	SINKS	BASIC	SINKING
BASIC	SIT	BASIC	SITS
BASIC	SITTING	BASIC	SKID
BASIC	SKIDS	BASIC	SKIDDING
BASIC	SKIP	BASIC	SKIPS
BASIC	SKIP'S	BASIC	SKIPPED
BASIC	SKIPPING	BASIC	SKUNK
BASIC	SKUNKS	BASIC	SKUNK'S
BASIC	SLAM	BASIC	SLAMS
BASIC	SLAMMED	BASIC	SLAMMING
BASIC	SLICK	BASIC	SLICKS
BASIC	SLICKED	BASIC	SLICKING
BASIC	SLID	BASIC	SLIP
BASIC	SLIPS	BASIC	SLIPPED
BASIC	SLIPPING	BASIC	SMASH
BASIC	SMASHES	BASIC	SMASHED
BASIC	SMASHING	BASIC	SNAP
BASIC	SNAPS	BASIC	SNAP'S
BASIC	SNAPPED	BASIC	SNAPPING
BASIC	SOCK	BASIC	SOCKS
BASIC	SOCKES	BASIC	SOCKING
BASIC	SPELL	BASIC	SPELLS
BASIC	SPELLED	BASIC	SPELLING
BASIC	SPIN	BASIC	SPINS
BASIC	SPINNING	BASIC	SPOT
BASIC	SPOTS	BASIC	SPOTTED
BASIC	SPOTTING	BASIC	STEEP
BASIC	STICK	BASIC	STICKS
BASIC	STICKING	BASIC	STILL
BASIC	STOP	BASIC	STOPS
BASIC	STOPPED	BASIC	STOPPING
BASIC	STUCK	BASIC	SUNK
BASIC	SWIM	BASIC	SWIMS
BASIC	SWIMMING	BASIC	TACK
BASIC	TACKS	BASIC	TACK'S
BASIC	TACKED	BASIC	TACKING
BASIC	TANK	BASIC	TANKS
BASIC	TANK'S	BASIC	TED

BASIC		TED'S
BASIC		TELLS
BASIC		TEN
BASIC		TENTS
BASIC		TENTED
BASIC		THANK
BASIC		THANKED
BASIC		THAT
BASIC		THEM
BASIC		THERE
BASIC		THINK
BASIC		THINKS
BASIC		THIS
BASIC		TINS
BASIC		TIPS
BASIC		TIPPED
BASIC		TO
BASIC		TOPS
BASIC		TOPPED
BASIC		TRACK
BASIC		TRACK'S
BASIC		TRACKING
BASIC		TRAPS
BASIC		TRAPPED
BASIC		TREE
BASIC		TREE'S
BASIC		TRICKS
BASIC		TRICKING
BASIC		TRIP
BASIC		TRIPPING
BASIC		TRUCKS
BASIC		TRUCKED
BASIC		TRUNK
BASIC		TRUNK'S
BASIC		TUBS
BASIC		TUT
BASIC		UP
BASIC		WANT
BASIC		WANTED
BASIC		WE
BASIC		WELLS
BASIC		WENT
BASIC		WETS
BASIC		WILL
BASIC		WINK
BASIC		WINK'S
BASIC		WINKING
BASIC		WISHES
BASIC		WISHES
BASIC		WITH
BASIC		YELLS
BASIC		YELLING
BASIC		YET
FIC	1	MOM
FIC	1	DAD
FIC	1	KIM
FIC	1	JACK
FIC	1	PEGGY

BASIC		TELL
BASIC		TELLING
BASIC		TENT
BASIC		TENT'S
BASIC		TENTING
BASIC		THANKS
BASIC		THANKING
BASIC		THE
BASIC		THEN
BASIC		THICK
BASIC		THINKING
BASIC		
BASIC		TIN
BASIC		TIP
BASIC		TIP'S
BASIC		TIPPING
BASIC		TOP
BASIC		TOP'S
BASIC		TOPPING
BASIC		TRACKS
BASIC		TRACKED
BASIC		TRAP
BASIC		TRAP'S
BASIC		TRAPPING
BASIC		TREES
BASIC		TRICK
BASIC		TRICKED
BASIC		TRIP
BASIC		TRIPPED
BASIC		TRUCK
BASIC		TRUCK'S
BASIC		TRUCKING
BASIC		TRUNKS
BASIC		TUB
BASIC		TUB'S
BASIC		TUT'S
BASIC		US
BASIC		WANTS
BASIC		WANTING
BASIC		WELL
BASIC		WELL'S
BASIC		WET
BASIC		WETTING
BASIC		WILL'S
BASIC		WINKS
BASIC		WINKED
BASIC		WISH
BASIC		WISHED
BASIC		WISHING
BASIC		YELL
BASIC		YELLED
BASIC		YES
BASIC		YOU
FIC	1	MOM'S
FIC	1	DAD'S
FIC	1	KIM'S
FIC	1	JACK'S
FIC	1	PEGGY'S

FIC	1	RUSS
FIC	2	BETH
FIC	2	BOB
FIC	2	STEVE
FIC	2	MARIA
FIC	2	ROSA
FIC	2	ROBERTO
FIC	3	SING
FIC	3	SINGING
FIC	3	SONGS
FIC	3	RANG
FIC	3	LONG
FIC	3	THING
FIC	3	BRINGS
FIC	3	KING
FIC	3	KING'S
FIC	3	RINGS
FIC	3	RINGING
FIC	4	CAVE
FIC	4	CAVE'S
FIC	4	GAME
FIC	4	GAME'S
FIC	4	SAFES
FIC	4	LATE
FIC	4	BANG
FIC	4	BANG'S
FIC	4	BANGING
FIC	4	TAKES
FIC	4	ATE
FIC	4	CAME
FIC	5	BIKE
FIC	5	BIKE'S
FIC	5	BIKING
FIC	5	SIDES
FIC	5	SIDED
FIC	5	TIRE
FIC	5	TIRE'S
FIC	5	TIRING
FIC	5	HIKES
FIC	5	HIKING
FIC	5	MILES
FIC	5	SHADE
FIC	5	SHADE'S
FIC	5	SHADING
FIC	5	LAKES
FIC	5	HIDE
FIC	5	HIDING
FIC	5	RIDES
FIC	5	MINE
FIC	5	MINE'S
FIC	5	MINING
FIC	5	FIRES
FIC	5	FIRED
FIC	5	TIME
FIC	5	TIMED
FIC	6	MORE
FIC	6	HOMES
FIC	6	HOMED

FIC	1	RUSS'S
FIC	2	BETH'S
FIC	2	BOB'S
FIC	2	STEVE'S
FIC	2	MARIA'S
FIC	2	ROSA'S
FIC	2	ROBERTO'S
FIC	3	SINGS
FIC	3	SONG
FIC	3	SONG'S
FIC	3	SANG
FIC	3	LONGING
FIC	3	BRING
FIC	3	BRINGING
FIC	3	KINGS
FIC	3	RING
FIC	3	RING'S
FIC	3	STRONG
FIC	4	CAVES
FIC	4	CAVED
FIC	4	GAMES
FIC	4	SAFE
FIC	4	SAFE'S
FIC	4	SAME
FIC	4	BANGS
FIC	4	BANGED
FIC	4	TAKE
FIC	4	TAKING
FIC	4	GAVE
FIC	4	MADE
FIC	5	BIKES
FIC	5	BIKED
FIC	5	SIDE
FIC	5	SIDE'S
FIC	5	SIDING
FIC	5	TIRES
FIC	5	TIRED
FIC	5	HIKE
FIC	5	HIKED
FIC	5	MILE
FIC	5	MILE'S
FIC	5	SHADES
FIC	5	SHADED
FIC	5	LAKE
FIC	5	LAKE'S
FIC	5	HIDES
FIC	5	RIDE
FIC	5	RIDING
FIC	5	MINES
FIC	5	MINED
FIC	5	FIRE
FIC	5	FIRE'S
FIC	5	FIRING
FIC	5	TIMES
FIC	5	TIMING
FIC	6	HOME
FIC	6	HOME'S
FIC	6	HOMING

FIC	6	BROKE
FIC	6	HOPES
FIC	6	HOPED
FIC	6	PIPE
FIC	6	PIPE'S
FIC	6	PIPING
FIC	6	ROPES
FIC	6	ROPED
FIC	6	WIPE
FIC	6	WIPE'S
FIC	6	WIPING
FIC	6	LIKES
FIC	6	LIKING
FIC	6	HOLES
FIC	6	SNORES
FIC	6	SNORING
FIC	6	NOTES
FIC	6	SMOKES
FIC	6	SMOKING
FIC	7	MICE
FIC	7	CAGE
FIC	7	CAGE'S
FIC	7	CAGING
FIC	7	RACES
FIC	7	RACED
FIC	7	RICE
FIC	7	POKES
FIC	7	POKED
FIC	7	STOVE
FIC	7	STOVE'S
FIC	7	SNEEZES
FIC	7	SNEEZING
FIC	7	FREEZES
FIC	7	MULE
FIC	7	MULE'S
FIC	7	BREEZES
FIC	7	BREEZING
FIC	7	JUNE'S
FIC	7	FLY'S
FIC	7	SKY
FIC	7	BY
FIC	7	TRY
FIC	7	ICE
FIC	7	ICED
FIC	7	PLACE
FIC	7	PLACED
FIC	7	HUGE
FIC	7	SPACES
FIC	7	SPACING
FIC	7	PAGES
FIC	8	PAGING
FIC	8	PARKS
FIC	8	PARKED
FIC	8	DARK
FIC	8	STARTS
FIC	8	STARTING
FIC	8	CARS
FIC	8	YARD

FIC	6	HOPE
FIC	6	HOPE'S
FIC	6	HOPING
FIC	6	PIPES
FIC	6	PIPED
FIC	6	ROPE
FIC	6	ROPE'S
FIC	6	ROPING
FIC	6	WIPES
FIC	6	WIPED
FIC	6	LIKE
FIC	6	LIKED
FIC	6	HOLE
FIC	6	SNORE
FIC	6	SNORED
FIC	6	NOTE
FIC	6	SMOKE
FIC	6	SMOKED
FIC	6	WOKE
FIC	7	NICE
FIC	7	CAGES
FIC	7	CAGED
FIC	7	RACE
FIC	7	RACE'S
FIC	7	RACING
FIC	7	POKE
FIC	7	POKE'S
FIC	7	POKING
FIC	7	STOVES
FIC	7	SNEEZE
FIC	7	SNEEZED
FIC	7	FREEZE
FIC	7	FREEZING
FIC	7	MULES
FIC	7	BREEZE
FIC	7	BREEZED
FIC	7	JUNE
FIC	7	FLY
FIC	7	FLYING
FIC	7	SKY'S
FIC	7	MY
FIC	7	TRYING
FIC	7	ICE'S
FIC	7	ICING
FIC	7	PLACES
FIC	7	PLACING
FIC	7	SPACE
FIC	7	SPACED
FIC	7	PAGE
FIC	7	PAGED
FIC	8	PARK
FIC	8	PARK'S
FIC	8	PARKING
FIC	8	START
FIC	8	STARTED
FIC	8	CAR
FIC	8	CAR'S
FIC	8	YARDS

FIC	8	YARD'S
FIC	8	FACES
FIC	8	FACED
FIC	8	FAR
FIC	8	JARS
FIC	8	JARRED
FIC	8	STAR
FIC	8	STAR'S
FIC	8	STARRING
FIC	8	CARTS
FIC	8	CARTED
FIC	8	HARD
FIC	9	CORNS
FIC	9	CORNED
FIC	9	FORKS
FIC	9	FORKED
FIC	9	SPORT
FIC	9	SPORT'S
FIC	9	SPORTING
FIC	9	FORT
FIC	9	FORT'S
FIC	9	PARTS
FIC	9	PARTING
FIC	9	MARKS
FIC	9	MARKED
FIC	9	BARK
FIC	9	BARK'S
FIC	9	BARKING
FIC	9	FOR
FIC	9	STORM'S
FIC	9	STORMED
FIC	9	SHORT
FIC	9	SHORTED
FIC	9	HORN
FIC	9	HORN'S
FIC	10	DIRT
FIC	10	SHIRTS
FIC	10	SHIRTING
FIC	10	HURTS
FIC	10	CURB
FIC	10	CURBS
FIC	10	CURBING
FIC	10	WORMS
FIC	10	WORMED
FIC	10	STORK
FIC	10	STORK'S
FIC	10	GIRLS
FIC	10	BIRD
FIC	10	BIRD'S
FIC	10	TURNS
FIC	10	TURNING
FIC	10	THIRD
FIC	11	INSIDE
FIC	11	JUMPROPE
FIC	11	JUMPROPE'S
FIC	11	JUMPROPING
FIC	11	FIREPLACE
FIC	11	FIREPLACE'S

FIC	8	FACE
FIC	8	FACE'S
FIC	8	FACING
FIC	8	JAR
FIC	8	JAR'S
FIC	8	JARRING
FIC	8	STARS
FIC	8	STARRED
FIC	8	CART
FIC	8	CART'S
FIC	8	CARTING
FIC	9	CORN
FIC	9	CORN'S
FIC	9	FORK
FIC	9	FORK'S
FIC	9	FORKING
FIC	9	SPORTS
FIC	9	SPORTED
FIC	9	TORN
FIC	9	FORTS
FIC	9	PART
FIC	9	PARTED
FIC	9	MARK
FIC	9	MARK'S
FIC	9	MARKING
FIC	9	BARKS
FIC	9	BARKED
FIC	9	OR
FIC	9	STORM
FIC	9	STORMS
FIC	9	STORMING
FIC	9	SHORTS
FIC	9	SHORTING
FIC	9	HORNS
FIC	9	HORNED
FIC	10	SHIRT
FIC	10	SHIRT'S
FIC	10	HURT
FIC	10	HURTING
FIC	10	CURB'S
FIC	10	CURBED
FIC	10	WORN
FIC	10	WORM'S
FIC	10	WORMING
FIC	10	STORKS
FIC	10	GIRL
FIC	10	GIRL'S
FIC	10	BIRDS
FIC	10	TURN
FIC	10	TURNED
FIC	10	FIRST
FIC	10	THIRDS
FIC	11	INSIDES
FIC	11	JUMPROPES
FIC	11	JUMPROPED
FIC	11	HOMEWORK
FIC	11	FIREPLACES
FIC	11	BEDTIME

FIC	11	BEDTIMES	FIC	11	WORK
FIC	11	WORKS	FIC	11	WORK'S
FIC	11	WORKED	FIC	11	WORKING
FIC	11	GIVE	FIC	11	GIVES
FIC	11	GIVING	FIC	11	HAVE
FIC	11	HAVING	FIC	11	BIG
FIC	11	DO	FIC	11	WAS
FIC	11	INTO	FIC	11	FISHPOND
FIC	11	FISHPOND	FIC	11	FISHPOND'S
FIC	11	UPSET	FIC	11	UPSETS
FIC	11	WITHOUT	FIC	11	UPON
FIC	12	LATER	FIC	12	LONGER
FIC	12	JUMPER	FIC	12	JUMPERS
FIC	12	JUMPER'S	FIC	12	SHORTER
FIC	12	STICKER	FIC	12	STICKERS
FIC	12	STICKER'S	FIC	12	STICKERED
FIC	12	OUTSIDE	FIC	12	THINKER
FIC	12	THINKERS	FIC	12	THINKER'S
FIC	12	SWIMMERS	FIC	12	SWIMMERS
FIC	12	SWIMMER'S	FIC	12	RIDER
FIC	12	RIDERS	FIC	12	RIDER'S
FIC	12	RUNNER	FIC	12	RUNNERS
FIC	12	RUNNER'S	FIC	12	KEEPER
FIC	12	KEEPERS	FIC	12	KEEPER'S
FIC	13	DIRTY	FIC	13	STORMY
FIC	13	TRICKY	FIC	13	MUDDY
FIC	13	HILLY	FIC	13	STEEPER
FIC	13	BADLY	FIC	13	GLADLY
FIC	13	HARDLY	FIC	13	SADLY
FIC	13	SHORTLY	FIC	13	DUSTY
FIC	13	FUNNY	FIC	13	LUCKY
FIC	13	ROCKY	FIC	13	SHORTY
FIC	13	SHORTY'S	FIC	14	LUNCH
FIC	14	LUNCHES	FIC	14	LUNCH'S
FIC	14	LUNCHED	FIC	14	LUNCHING
FIC	14	PORCH	FIC	14	PORCHES
FIC	14	PORCH'S	FIC	14	BENCH
FIC	14	BENCHES	FIC	14	BENCH'S
FIC	14	CHIN	FIC	14	CHINS
FIC	14	CHIN'S	FIC	14	CHINNED
FIC	14	CHINNING	FIC	14	CATCHUP
FIC	14	STICKY	FIC	14	WITCH
FIC	14	WITCHES	FIC	14	WITCH'S
FIC	14	CATCH	FIC	14	CATCHES
FIC	14	CATCHING	FIC	14	PINCH
FIC	14	PINCHES	FIC	14	PINCHED
FIC	14	PINCHING	FIC	14	MUCH
FIC	14	SANDWICH	FIC	14	SANDWICHES
FIC	14	SANDWICHED	FIC	14	SANDWICHING
FIC	15	QUIT	FIC	15	QUITS
FIC	15	QUITTING	FIC	15	QUACK
FIC	15	QUACKS	FIC	15	QUACK'S
FIC	15	QUACKED	FIC	15	QUACKING
FIC	15	WHITE	FIC	15	WHITES
FIC	15	WHITE'S	FIC	15	WHILE
FIC	15	WHY	FIC	15	SCRATCH
FIC	15	SCRATCHES	FIC	15	SCRATCHED
FIC	15	SCRATCHING	FIC	15	QUICK

FIC	15	QUICKSAND
FIC	15	QUEENS
FIC	15	WHICH
FIC	16	TODAY
FIC	16	HAY'S
FIC	16	SUNDAY
FIC	16	SAY
FIC	16	SAYING
FIC	16	WHEEL
FIC	16	WHEELED
FIC	16	WHAT
FIC	16	HER
FIC	16	THAN
FIC	16	WAITS
FIC	16	WAITED
FIC	16	STAYS
FIC	16	STAYING
FIC	16	WAYS
FIC	16	PAYS
FIC	16	MAY
FIC	16	DAYS
FIC	16	SUNDAY'S
FIC	17	SCREAM
FIC	17	SCREAMED
FIC	17	TEACHER
FIC	17	TEACHER'S
FIC	17	SEATS
FIC	17	SEATING
FIC	17	EACH
FIC	17	SQUEAKS
FIC	17	SQUEAKED
FIC	17	GRAY
FIC	17	GRAY'S
FIC	17	GRAYING
FIC	17	EATS
FIC	17	SEA
FIC	17	SEA'S
FIC	17	HEARS
FIC	17	NEAR
FIC	17	NEARED
FIC	17	BEAST
FIC	18	NOONS
FIC	18	ZOO
FIC	18	ZOO'S
FIC	18	POOLS
FIC	18	POOLED
FIC	18	TOO
FIC	18	FOODS
FIC	18	LEAVE
FIC	18	LEAVE'S
FIC	18	REACH
FIC	18	REACHED
FIC	18	SEAL
FIC	18	SEAL'S
FIC	18	SEALING
FIC	18	BROOM
FIC	18	BROOM'S
FIC	18	ROOMS

FIC	15	QUEEN
FIC	15	QUEEN'S
FIC	15	WHEN
FIC	16	HAY
FIC	16	MAYBE
FIC	16	SUNDAYS
FIC	16	SAYS
FIC	16	WHEELS
FIC	16	WHEEL'S
FIC	16	WHEELING
FIC	16	WHO
FIC	16	HERS
FIC	16	WAIT
FIC	16	WAITING
FIC	16	STAY
FIC	16	STAYED
FIC	16	WAY
FIC	16	PAY
FIC	16	PAYING
FIC	16	DAY
FIC	16	DAY'S
FIC	17	BEAST'S
FIC	17	SCREAMS
FIC	17	SCREAMING
FIC	17	TEACHERS
FIC	17	SEAT
FIC	17	SEAT'S
FIC	17	SEATED
FIC	17	SQUEAK
FIC	17	SQUEAK'S
FIC	17	SQUEAKING
FIC	17	GRAYS
FIC	17	GRAYED
FIC	17	EAT
FIC	17	EATING
FIC	17	SEAS
FIC	17	HEAR
FIC	17	HEARING
FIC	17	NEARS
FIC	17	NEARING
FIC	17	BEASTS
FIC	18	NOON'S
FIC	18	ZOOS
FIC	18	POOL
FIC	18	POOL'S
FIC	18	POOLING
FIC	18	FOOD
FIC	18	FOOD'S
FIC	18	LEAVES
FIC	18	LEAVING
FIC	18	REACHES
FIC	18	REACHING
FIC	18	SEALS
FIC	18	SEALED
FIC	18	SOON
FIC	18	BROOMS
FIC	18	ROOM
FIC	18	ROOM'S

FIC	18	ROOMED	FIC	18	ROOMING
FIC	18	POOR	FIC	18	MOON
FIC	18	MOONS	FIC	18	MOON'S
FIC	18	NOON	FIC	19	BOWL'S
FIC	19	GROWN	FIC	19	GROWNS
FIC	19	GROWNED	FIC	19	GROWNING
FIC	19	BLOWN	FIC	19	SHOW
FIC	19	SHOWS	FIC	19	SHOW'S
FIC	19	SHOWED	FIC	19	SHOWING
FIC	19	THROW	FIC	19	THROWS
FIC	19	THROWING	FIC	19	BOWL
FIC	19	BOWLS	FIC	19	BOWLED
FIC	19	BOWLING	FIC	19	SCOOP
FIC	19	SCOOPS	FIC	19	SCOOP'S
FIC	19	SCOOPED	FIC	19	SCOOPING
FIC	19	COOL	FIC	19	COOLS
FIC	19	COOLED	FIC	19	COOLING
FIC	19	ROOF	FIC	19	ROOFS
FIC	19	ROOF'S	FIC	19	ROOFED
FIC	19	ROOFING	FIC	19	IF
FIC	19	WERE	FIC	19	HERE
FIC	19	THEY	FIC	19	YOUR
FIC	19	YOURS	FIC	19	SNOWS
FIC	19	SNOW	FIC	19	SNOWED
FIC	19	SNOWING	FIC	19	SLOWLY
FIC	19	WINDOW	FIC	19	WINDOWS
FIC	19	WINDOW'S	FIC	19	GLOW
FIC	19	GLOWS	FIC	19	GLOWED
FIC	19	GLOWING	FIC	19	FOLLOW
FIC	19	FOLLOWS	FIC	19	FOLLOWED
FIC	19	FOLLOWING	FIC	20	SCOLD
FIC	20	SCOLDS	FIC	20	SCOLDED
FIC	20	SCOLDING	FIC	20	GOLDFISH
FIC	20	GOLDFISHES	FIC	20	GOLDFISH'S
FIC	20	HOLD	FIC	20	HOLDS
FIC	20	HOLLING	FIC	20	SOLD
FIC	20	ROLL	FIC	20	ROLLS
FIC	20	ROLL'S	FIC	20	ROLLED
FIC	20	ROLLING	FIC	20	BORROW
FIC	20	BORROWS	FIC	20	BORROWED
FIC	20	BORROWING	FIC	20	FALL
FIC	20	FALLS	FIC	20	FALLING
FIC	20	BALL	FIC	20	BALLS
FIC	20	BALL'S	FIC	20	SMALL
FIC	20	TALL	FIC	20	CALL
FIC	20	CALLS	FIC	20	CALLED
FIC	20	CALLING	FIC	20	OLD
FIC	20	GOLD	FIC	20	COLD
FIC	20	TROLL	FIC	20	TROLLS
FIC	20	TROLL'S	FIC	20	TOLD
FIC	21	HEAD	FIC	21	HEADS
FIC	21	HEAD'S	FIC	21	HEADED
FIC	21	HEADING	FIC	21	INSTEAD
FIC	21	WEAR	FIC	21	WEARS
FIC	21	WEARING	FIC	21	FOREHEAD
FIC	21	FOREHEADS	FIC	21	FOREHEAD'S
FIC	21	TEAR	FIC	21	TEARS
FIC	21	TEARING	FIC	21	FOLD

FIC	21	FOLDS
FIC	21	FOLDING
FIC	21	AS
FIC	21	WALKS
FIC	21	WALKING
FIC	21	GIANT
FIC	21	GIANT'S
FIC	21	FINDS
FIC	21	KIND
FIC	21	MIND
FIC	21	MINDED
FIC	21	CHILD
FIC	21	BREAD
FIC	21	BREAD'S
FIC	21	BREADING
FIC	21	SPREADS
FIC	21	THREAD
FIC	21	THREADED
FIC	21	BEAR
FIC	21	BEARS
FIC	21	PEAR'S
FIC	21	PEAR
FIC	22	COOK'S
FIC	22	COOKS
FIC	22	COOKED
FIC	22	BOOK
FIC	22	BOOK'S
FIC	22	BOOKING
FIC	22	WOOD
FIC	22	WOOD'S
FIC	22	WOODING
FIC	22	READ
FIC	22	READING
FIC	22	GOOD
FIC	22	SHOOK
FIC	22	FOOTED
FIC	22	LOOK
FIC	23	GROWL
FIC	23	GROWLS
FIC	23	GROWLING
FIC	23	CLOWNS
FIC	23	CLOWNED
FIC	23	BOW
FIC	23	BOWED
FIC	23	CROWD
FIC	23	CROWD'S
FIC	23	CROWDING
FIC	23	FROWNS
FIC	23	FROWNING
FIC	23	GOOD-BYS
FIC	23	SO
FIC	23	WOMAN
FIC	23	SOUP
FIC	23	HOW
FIC	23	COW
FIC	23	COW'S
FIC	23	TOWNS
FIC	23	CROWN

FIC	21	FOLDED
FIC	21	WHERE
FIC	21	WALK
FIC	21	WALKED
FIC	21	MAGIC
FIC	21	GIANTS
FIC	21	FIND
FIC	21	FINDING
FIC	21	KINDS
FIC	21	MINDS
FIC	21	MINDING
FIC	21	WILD
FIC	21	BREADS
FIC	21	BREADED
FIC	21	SPREAD
FIC	21	SPREADING
FIC	21	THREADS
FIC	21	THREADING
FIC	21	BEARING
FIC	21	BEAR'S
FIC	21	PEARS
FIC	22	COOK
FIC	22	LOOKED
FIC	22	COOKING
FIC	22	BOOKS
FIC	22	BOOKED
FIC	22	STOOD
FIC	22	WOODS
FIC	22	WOODED
FIC	22	FIREWOOD
FIC	22	READS
FIC	22	TOOK
FIC	22	GOODS
FIC	22	FOOT
FIC	22	FOOTING
FIC	22	LOOKS
FIC	22	LOOKING
FIC	23	CROWNING
FIC	23	GROWLED
FIC	23	CLOWN
FIC	23	CLOWN'S
FIC	23	CLOWNING
FIC	23	BOWS
FIC	23	BOWING
FIC	23	CROWDS
FIC	23	CROWDED
FIC	23	FROWN
FIC	23	FROWNED
FIC	23	GOOD-BY
FIC	23	DOES
FIC	23	NO
FIC	23	WOMAN'S
FIC	23	SOUPS
FIC	23	NOW
FIC	23	COWS
FIC	23	TOWN
FIC	23	TOWN'S
FIC	23	CROWNS

FIC	23	CROWN'S	FIC	23	CROWNED
FIC	24	BASHFUL	FIC	24	BASHFUL'S
FIC	24	BULLDOG	FIC	24	BULLDOGS
FIC	24	BULLDOG'S	FIC	24	BUSH
FIC	24	BUSHES	FIC	24	BUSH'S
FIC	24	BUSHED	FIC	24	BUSHING
FIC	24	FULL	FIC	24	BROWN
FIC	24	BROWNS	FIC	24	BROWN'S
FIC	24	BROWNED	FIC	24	BROWNING
FIC	24	SOME	FIC	24	COME
FIC	24	COMES	FIC	24	COMING
FIC	24	SON	FIC	24	SONS
FIC	24	SON'S	FIC	24	NOTHING
FIC	24	DONE	FIC	24	PUSH
FIC	24	PUSHES	FIC	24	PUSHED
FIC	24	PUSHING	FIC	24	BULL
FIC	24	BULLS	FIC	24	BULL'S
FIC	24	PLAYFUL	FIC	24	CAREFUL
FIC	24	PULLS	FIC	24	PULLS
FIC	24	PULLED	FIC	24	PULLING
FIC	24	HANDFUL	FIC	25	HUNGRY
FIC	25	HURRY	FIC	25	HAPPY
FIC	25	HAPPY'S	FIC	25	CANDY
FIC	25	CANDY'S	FIC	25	TAFFY
FIC	25	TAFFY'S	FIC	25	PLENTY
FIC	25	SPOONFUL	FIC	25	EMPTY
FIC	25	EMPTYING	FIC	25	DIZZY
FIC	25	JOLLY	FIC	25	READY
FIC	26	LITTLE'S	FIC	26	BUBBLING
FIC	26	LITTLE	FIC	26	UNCLES
FIC	26	UNCLE	FIC	26	RATTLE
FIC	26	UNCLE'S	FIC	26	RATTLE'S
FIC	26	RATTLES	FIC	26	RATTLING
FIC	26	RATTLED	FIC	26	PUZZLE
FIC	26	STORY	FIC	26	STORY'S
FIC	26	BEEN	FIC	26	MANY
FIC	26	VERY	FIC	26	ONE
FIC	26	ONCE	FIC	26	JUNGLE
FIC	26	JUNGLES	FIC	26	JUNGLE'S
FIC	26	BOTTLE	FIC	26	BOTTLES
FIC	26	BOTTLING	FIC	26	BOTTLED
FIC	26	BOTTLE'S	FIC	26	GENTLE
FIC	26	GENTLES	FIC	26	GENTLED
FIC	26	GENTLING	FIC	26	MIDDLE
FIC	26	BUBBLE	FIC	26	BUBBLES
FIC	26	BUBBLE'S	FIC	26	BUBBLED
FIC	26	SIMPLE	FIC	27	HORSE
FIC	27	HORSES	FIC	27	HORSE'S
FIC	27	EDGE	FIC	27	EDGES
FIC	27	EDGE'S	FIC	27	EDGED
FIC	27	EDGING	FIC	27	FENCE
FIC	27	FENCES	FIC	27	FENCE'S
FIC	27	FENCED	FIC	27	FENCING
FIC	27	SINCE	FIC	27	FUDGE
FIC	27	FUDGED	FIC	27	FUDGING
FIC	27	SADDLE	FIC	27	SADDLES
FIC	27	SADDLE'S	FIC	27	SADDLED
FIC	27	SADDLING	FIC	27	PUDDLE

FIC	27	PUDDLES
FIC	27	PUDDLED
FIC	27	PURSE
FIC	27	PURSE'S
FIC	27	PRINCES
FIC	27	ELSE
FIC	27	DANCES
FIC	27	DANCED
FIC	27	BRIDGE
FIC	27	BRIDGE'S
FIC	27	BRIDGING
FIC	28	BASKETS
FIC	28	COVER
FIC	28	COVER'S
FIC	28	COVERING
FIC	28	WONDER
FIC	28	WONDERED
FIC	28	KITTENS
FIC	28	LARGE
FIC	28	THUNDER
FIC	28	THUNDERED
FIC	28	FOREST
FIC	28	FOREST'S
FIC	28	HAPPENS
FIC	28	HAPPENING
FIC	28	ENTERS
FIC	28	ENTERING
FIC	29	BANDIT
FIC	29	BANDIT'S
FIC	29	CANYONS
FIC	29	PENCIL
FIC	29	PENCIL'S
FIC	29	PENCILING
FIC	29	BOTTOM
FIC	29	PUMPKIN
FIC	29	PUMPKIN'S
FIC	29	GOBLINS
FIC	29	MIRROR
FIC	29	MIRROR'S
FIC	29	MIRRORING
FIC	29	RABBITS
FIC	29	PERSON
FIC	29	PERSON'S
FIC	29	DARKNESS
FIC	29	ANSWER
FIC	29	ANSWERED
FIC	29	FRIEND
FIC	29	FRIEND'S
FIC	29	SAWS
FIC	29	SAWED
FIC	29	BOY
FIC	30	ALONG
FIC	30	RASCAL
FIC	30	RASCAL'S
FIC	30	BREAKFASTS
FIC	30	TENNIS
FIC	30	BALLOON
FIC	30	BALLOON'S

FIC	27	PUDDLE'S
FIC	27	PUDDLING
FIC	27	PURSES
FIC	27	PRINCE
FIC	27	PRINCE'S
FIC	27	DANCE
FIC	27	DANCE'S
FIC	27	DANCING
FIC	27	BRIDGES
FIC	27	BRIDGED
FIC	28	BASKET
FIC	28	BASKET'S
FIC	28	COVERS
FIC	28	COVERED
FIC	28	UNDER
FIC	28	WONDERS
FIC	28	WONDERING
FIC	28	KITTEN'S
FIC	28	AFTER
FIC	28	THUNDERS
FIC	28	THUNDERING
FIC	28	FORESTS
FIC	28	HAPPEN
FIC	28	HAPPENED
FIC	28	ENTER
FIC	28	ENTERED
FIC	29	BOY'S
FIC	29	BANDITS
FIC	29	CANYON
FIC	29	CANYON'S
FIC	29	PENCILS
FIC	29	PENCILED
FIC	29	STUBBORN
FIC	29	BOTTOMS
FIC	29	PUMPKINS
FIC	29	GOBLIN
FIC	29	GOBLIN'S
FIC	29	MIRRORS
FIC	29	MIRRORED
FIC	29	RABBIT
FIC	29	RABBIT'S
FIC	29	PERSONS
FIC	29	SUDDEN
FIC	29	GONE
FIC	29	ANSWERS
FIC	29	ANSWERING
FIC	29	FRIENDS
FIC	29	SAW
FIC	29	SAW'S
FIC	29	SAWING
FIC	29	BOYS
FIC	30	ALIKE
FIC	30	RASCALS
FIC	30	BREAKFAST
FIC	30	FIREMAN
FIC	30	AWAY
FIC	30	BALLOONS
FIC	30	ALONE

FIC	30	ABOVE	FIC	30	ACROSS
FIC	30	DOOR	FIC	30	DOORS
FIC	30	DOOR'S	FIC	30	FLOOR
FIC	30	FLOORS	FIC	30	FLOOR'S
FIC	30	FLOOD	FIC	30	FLOODS
FIC	30	FLOOD'S	FIC	30	FLOODED
FIC	30	FLOODING	FIC	30	DWARF
FIC	30	DWARFS	FIC	30	DWARF'S
FIC	30	DWARFED	FIC	30	CASTLE
FIC	30	CASTLES	FIC	30	CASTLE'S
FAN	7	THING	FAN	7	BRING
FAN	7	BRINGS	FAN	7	BRINGING
FAN	7	KING	FAN	7	KINGS
FAN	7	KING'S	FAN	7	RING
FAN	7	RINGS	FAN	7	RING'S
FAN	7	RINGING	FAN	7	STRONG
FAN	8	TAKE	FAN	8	TAKES
FAN	8	TAKING	FAN	8	ATE
FAN	8	CAME	FAN	8	GAVE
FAN	8	MADE	FAN	9	HIDE
FAN	9	HIDES	FAN	9	HIDING
FAN	9	RIDE	FAN	9	RIDES
FAN	9	RIDING	FAN	9	MINE
FAN	9	MINES	FAN	9	MINE'S
FAN	9	MINED	FAN	9	MINING
FAN	9	FIRE	FAN	9	FIRES
FAN	9	FIRE'S	FAN	9	FIRED
FAN	9	FIRING	FAN	9	TIME
FAN	9	TIMES	FAN	9	TIMED
FAN	9	TIMING	FAN	10	HOLE
FAN	10	HOLES	FAN	10	SNORE
FAN	10	SNORES	FAN	10	SNORED
FAN	10	SNORING	FAN	10	NOTE
FAN	10	NOTES	FAN	10	SMOKE
FAN	10	SMOKES	FAN	10	SMOKED
FAN	10	SMOKING	FAN	10	WOKE
FAN	11	SNEEZE	FAN	11	SNEEZES
FAN	11	SNEEZED	FAN	11	SNEEZING
FAN	11	FREEZE	FAN	11	FREEZES
FAN	11	FREEZING	FAN	11	MULE
FAN	11	MULES	FAN	11	MULE'S
FAN	11	BREEZE	FAN	11	BREEZES
FAN	11	BREEZED	FAN	11	BREEZING
FAN	11	JUNE	FAN	11	JUNE'S
FAN	12	FLY	FAN	12	FLY'S
FAN	12	FLYING	FAN	12	SKY
FAN	12	SKIES	FAN	12	BY
FAN	12	MY	FAN	12	TRY
FAN	12	TRYING	FAN	13	ICE
FAN	13	ICES	FAN	13	ICED
FAN	13	ICING	FAN	13	PLACE
FAN	13	PLACES	FAN	13	PLACED
FAN	13	PLACING	FAN	13	HUGE
FAN	13	SPACE	FAN	13	SPACES
FAN	13	SPACED	FAN	13	SPACING
FAN	13	PAGE	FAN	13	PAGES
FAN	13	PAGED	FAN	13	PAGING
FAN	14	FAR	FAN	14	JAR

FAN	14	JARS	FAN	14	JAR'S
FAN	14	JARRED	FAN	14	JARRING
FAN	14	STAR	FAN	14	STARS
FAN	14	STAR'S	FAN	14	STARRED
FAN	14	STARRING	FAN	14	CART
FAN	14	CARTS	FAN	14	CART'S
FAN	14	CARTED	FAN	14	CARTING
FAN	14	HARD	FAN	15	OR
FAN	15	FOR	FAN	15	STORM
FAN	15	STORMS	FAN	15	STORM'S
FAN	15	STORMED	FAN	15	STORMING
FAN	15	SHORT	FAN	15	SHORTS
FAN	15	SHORTED	FAN	15	SHORTING
FAN	15	HORN	FAN	15	HORNS
FAN	15	HORN'S	FAN	15	HORNED
FAN	16	GIRL	FAN	16	GIRLS
FAN	16	GIRL'S	FAN	16	BIRD
FAN	16	BIRDS	FAN	16	BIRD'S
FAN	16	TURN	FAN	16	TURNS
FAN	16	TURNED	FAN	16	TURNING
FAN	16	FIRST	FAN	16	THIRD
FAN	16	THIRDS	FAN	17	GIVE
FAN	17	GIVES	FAN	17	GIVING
FAN	17	HAVE	FAN	17	HAVING
FAN	17	BIG	FAN	17	DO
FAN	17	WAS	FAN	18	INTO
FAN	18	FISHPOND	FAN	18	FISHPONDS
FAN	18	UPSET	FAN	18	UPSETS
FAN	18	WITHOUT	FAN	18	UPON
FAN	19	THINKER	FAN	19	THINKERS
FAN	19	THINKER'S	FAN	19	SWIMMER
FAN	19	SWIMMERS	FAN	19	SWIMMER'S
FAN	19	RIDER	FAN	19	RIDERS
FAN	19	RIDER'S	FAN	19	RUNNER
FAN	19	RUNNERS	FAN	19	RUNNER'S
FAN	19	KEEPER	FAN	19	KEEPERS
FAN	19	KEEPER'S	FAN	20	BADLY
FAN	20	GLADLY	FAN	20	HARDLY
FAN	20	SADLY	FAN	20	SHORTLY
FAN	21	DUSTY	FAN	21	FUNNY
FAN	21	LUCKY	FAN	21	ROCKY
FAN	21	SHORTY	FAN	21	SHORTY'S
FAN	22	WITCH	FAN	22	WITCHES
FAN	22	WITCH'S	FAN	22	CATCH
FAN	22	CATCHES	FAN	22	CATCHING
FAN	22	PINCH	FAN	22	PINCHES
FAN	22	PINCHED	FAN	22	PINCHING
FAN	22	MUCH	FAN	22	SANDWICH
FAN	22	SANDWICHES	FAN	22	SANDWICHED
FAN	22	SANDWICHING	FAN	23	QUICK
FAN	23	QUICKSAND	FAN	23	QUEEN
FAN	23	QUEENS	FAN	23	QUEEN'S
FAN	23	WHICH	FAN	23	WHEN
FAN	24	WHAT	FAN	24	WHO
FAN	24	HER	FAN	24	HERS
FAN	24	THAN	FAN	24	WAIT
FAN	24	WAITS	FAN	24	WAITED
FAN	24	WAITING	FAN	25	STAY

FAN	25	STAYS	FAN	25	STAYED
FAN	25	STAYING	FAN	25	WAY
FAN	25	WAYS	FAN	25	PAY
FAN	25	PAYS	FAN	25	PAYING
FAN	25	MAY	FAN	25	DAY
FAN	25	DAYS	FAN	25	DAY'S
FAN	26	EAT	FAN	26	EATS
FAN	26	EATING	FAN	26	SEA
FAN	26	SEAS	FAN	26	SEA'S
FAN	26	HEAR	FAN	26	HEARS
FAN	26	HEARING	FAN	26	NEAR
FAN	26	NEARS	FAN	26	NEARED
FAN	26	NEARING	FAN	26	BEAST
FAN	26	BEASTS	FAN	26	BEAST'S
FAN	27	SOON	FAN	27	BROOM
FAN	27	BROOMS	FAN	27	BROOM'S
FAN	27	ROOM	FAN	27	ROOMS
FAN	27	ROOM'S	FAN	27	ROOMED
FAN	27	ROOMMING	FAN	27	POOR
FAN	27	MOON	FAN	27	MOONS
FAN	27	MOON'S	FAN	28	IF
FAN	28	WERE	FAN	28	HERE
FAN	28	THEY	FAN	28	YOUR
FAN	28	YOURS	FAN	29	SNOW
FAN	29	SNOWS	FAN	29	SNOWED
FAN	29	SNOWING	FAN	29	SLOWLY
FAN	29	WINDOW	FAN	29	WINDOWS
FAN	29	WINDOW'S	FAN	29	GLOW
FAN	29	GLOWS	FAN	29	GLOWED
FAN	29	GLOWING	FAN	29	FOLLOW
FAN	29	FOLLOWS	FAN	29	FOLLOWED
FAN	29	FOLLOWING	FAN	30	FALL
FAN	30	FALLS	FAN	30	FALLING
FAN	30	BALL	FAN	30	BALLS
FAN	30	BALL'S	FAN	30	SMALL
FAN	30	TALL	FAN	30	CALL
FAN	30	CALLS	FAN	30	CALLED
FAN	30	CALLING	FAN	31	OLD
FAN	31	GOLD	FAN	31	COLD
FAN	31	TROLL	FAN	31	TROLLS
FAN	31	TROLL'S	FAN	31	TOLD
FAN	32	WHERE	FAN	32	AS
FAN	32	WALK	FAN	32	WALKS
FAN	32	WALKED	FAN	32	WALKING
FAN	32	MAGIC	FAN	32	GIANT
FAN	32	GIANTS	FAN	32	GIANT'S
FAN	33	FIND	FAN	33	FINDS
FAN	33	FINDING	FAN	33	KIND
FAN	33	KINDS	FAN	33	MIND
FAN	33	MINDS	FAN	33	MINDED
FAN	33	MINDING	FAN	33	CHILD
FAN	33	WILD	FAN	34	BREAD
FAN	34	BREADS	FAN	34	BREAD'S
FAN	34	BREADED	FAN	34	BREADING
FAN	34	SPREAD	FAN	34	SPREADS
FAN	34	SPREADING	FAN	34	THREAD
FAN	34	THREADS	FAN	34	THREADED
FAN	34	THREADING	FAN	34	BEAR

FAN	34	BEARS	FAN	34	BEAR'S
FAN	34	BEARING	FAN	34	PEAR
FAN	34	PEARS	FAN	34	PEAR'S
FAN	35	TOOK	FAN	35	GOOD
FAN	35	GOODS	FAN	35	SHOOK
FAN	35	FOOT	FAN	35	FOOTED
FAN	35	FOOTING	FAN	35	LOOK
FAN	35	LOOKS	FAN	35	LOOKED
FAN	35	LOOKING	FAN	36	DOES
FAN	36	SO	FAN	36	NO
FAN	36	WOMAN	FAN	36	WOMAN'S
FAN	36	SOUP	FAN	36	SOUPS
FAN	37	HOW	FAN	37	NOW
FAN	37	COW	FAN	37	COWS
FAN	37	COW'S	FAN	37	TOWN
FAN	37	TOWNS	FAN	37	TOWN'S
FAN	37	CROWN	FAN	37	CROWNS
FAN	37	CROWN'S	FAN	37	CROWNED
FAN	37	CROWNING	FAN	38	SOME
FAN	38	COME	FAN	38	COMES
FAN	38	COMING	FAN	38	SON
FAN	38	SONS	FAN	38	SON'S
FAN	38	NOTHING	FAN	38	DONE
FAN	39	PUSH	FAN	39	PUSHES
FAN	39	PUSHED	FAN	39	PUSHING
FAN	39	BULL	FAN	39	BULLS
FAN	39	BULL'S	FAN	39	PLAYFUL
FAN	39	CAREFUL	FAN	39	PULL
FAN	39	PULLS	FAN	39	PULLED
FAN	39	PULLING	FAN	40	EMPTY
FAN	40	EMPTYING	FAN	40	JOLLY
FAN	40	DIZZY	FAN	40	HUNGRY
FAN	40	READY	FAN	41	BEEN
FAN	41	MANY	FAN	41	VERY
FAN	41	ONE	FAN	41	ONCE
FAN	42	JUNGLE	FAN	42	JUNGLES
FAN	42	JUNGLE'S	FAN	42	BOTTLE
FAN	42	BOTTLES	FAN	42	BOTTLE'S
FAN	42	BOTTLED	FAN	42	BOTTLING
FAN	42	GENTLE	FAN	42	GENTLES
FAN	42	GENTLED	FAN	42	GENTLING
FAN	42	MIDDLE	FAN	42	BUBBLE
FAN	42	BUBBLES	FAN	42	BUBBLE'S
FAN	42	BUBBLED	FAN	42	BUBBLING
FAN	43	PURSE	FAN	43	PURSES
FAN	43	PURSE'S	FAN	43	PRINCE
FAN	43	PRINCES	FAN	43	PRINCE'S
FAN	43	ELSE	FAN	43	DANCE
FAN	43	DANCES	FAN	43	DANCED
FAN	43	DANCING	FAN	43	BRIDGE
FAN	43	BRIDGES	FAN	43	BRIDGE'S
FAN	43	BRIDGED	FAN	43	BRIDGING
FAN	44	AFTER	FAN	44	THUNDER
FAN	44	THUNDERS	FAN	44	THUNDERED
FAN	44	THUNDERING	FAN	44	FOREST
FAN	44	FORESTS	FAN	44	FOREST'S
FAN	44	HAPPEN	FAN	44	HAPPENS
FAN	44	HAPPENED	FAN	44	HAPPENING

FAN	44	ENTER	FAN	44	ENTERS
FAN	44	ENTERED	FAN	44	ENTERING
FAN	45	GONE	FAN	45	ANSWER
FAN	45	ANSWERS	FAN	45	ANSWERED
FAN	45	ANSWERING	FAN	45	FRIEND
FAN	45	FRIENDS	FAN	45	FRIEND'S
FAN	45	SAW	FAN	45	SAWS
FAN	45	SAW'S	FAN	45	SAWED
FAN	45	SAWING	FAN	45	BOY
FAN	45	BOYS	FAN	45	BOY'S
FAN	46	BANDIT	FAN	46	BANDITS
FAN	46	BANDIT'S	FAN	46	CANYON
FAN	46	CANYONS	FAN	46	CANYON'S
FAN	46	PENCIL	FAN	46	PENCILS
FAN	46	PENCIL'S	FAN	46	PENCILED
FAN	46	PENCILING	FAN	46	STUBBORN
FAN	46	BOTTOM	FAN	46	BOTTOMS
FAN	47	AWAY	FAN	47	BALLOON
FAN	47	BALLOONS	FAN	47	BALLOON'S
FAN	47	ALONE	FAN	47	ABOVE
FAN	47	ACROSS	FAN	48	DOOR
FAN	48	DOORS	FAN	48	DOOR'S
FAN	48	FLOOR	FAN	48	FLOORS
FAN	48	FLOOR'S	FAN	48	FLOOD
FAN	48	FLOODS	FAN	48	FLOOD'S
FAN	48	FLOODED	FAN	48	FLOODING
FAN	48	DWARF	FAN	48	DWARFS
FAN	48	DWARF'S	FAN	48	DWARFED
FAN	48	CASTLE	FAN	48	CASTLES
FAN	48	CASTLE'S	FAN		

END

BAS ANI STORY # 5

A PLAY

TIP. IN THIS PLAY, I AM ME, PAT. IN THIS PLAY, I AM ME, BUD. SEE THIS MAS
ON ME. I AM A TREE, PAT. WE WILL CAMP NEXT TO THIS POND, TIP. TIP. YES, P
. IT IS FUN TO CAMP OUT. TIP, HELP ME WITH THE TENT, PAT. WE NEED A LOG T
KEEP IT UP, PAT. SEE THE TREES, WE CAN GET ALL THE LOGS WE WANT. TIP. WE
LL CUT THIS TREE FOR LOGS, TIP. PAT JUST WENT UP THE TREE TO CUT IT. TREE
STOP. STOP. YOU MUST NOT CUT ME DOWN. PAT. DID THIS TREE ASK ME TO STOP.
EE. YES, I DID. TIP. DO NOT STOP. PAT. WE NEED LOGS. TREE. I WILL TRICK T
M. I WILL JUMP UP AND DOWN. PAT. HELP. HELP. KEEP STILL, TREE. TIP. PAT F
L FROM THE TREE. HE FELL IN THE POND. PAT. I AM ALL WET, TIP. HELP ME OUT
TIP. I CAN NOT SWIM. PAT. I CAN NOT HELP YOU. TREE. I WILL HELP YOU OUT.
T. PAT. THE TREE HAS ME. TREE. I WILL PUT YOU ON THIS ROCK. PAT. I AM GLA
TO BE OUT OF THIS POND. I AM GLAD TO BE BACK ON LAND. TIP. TREE. WE WILL
T CUT YOU DOWN.

FICTION STORY #1

JACK GETS STUCK

KIM AND JACK SAT ON TOP OF A TRUNK. MOM AND DAD SAT NEXT TO THEM. LET US
OUT AND PLAY, SAID JACK TO KIM. WE MUST ASK MOM AND DAD, SAID KIM. MOM,
D. CAN WE GO OUT TO PLAY. ASKED KIM. YES. YOU CAN. SAID DAD. KIM AND JACK
ENT OUT. PEGGY AND RUSS SAT NEXT TO A TREE. PLAY WITH US, SAID RUSS. PEGG
WANTED TO BE IT. RUSS, KIM, AND JACK ALL HID. RUSS HID IN BACK OF A TREE
D KIM HID IN BACK OF A BOX. JACK PICKED THE CUT GRASS OUT OF A CAN AND JU
ED IN. PEGGY WILL NOT SEE ME IN THIS CAN. JACK SAID. PEGGY SPOTTED RUSS I
THE GRASS NEXT TO THE TREE. RUSS, I SEE YOU, YELLED PEGGY. THEN PEGGY SPO
ED KIM IN BACK OF THE BOX. KIM, I SEE YOU, YELLED PEGGY. PEGGY DID NOT SE

JACK. RUSS AND KIM HELPED PEGGY. HE IS NOT IN BACK OF THIS LOG, SAID KIM.
EGGY JUMPED UP. I THINK I SEE HIM, SHE SAID. THE CUT GRASS IN OUT OF THAT
AN. I THINK HE HID IN THERE. PEGGY WENT TO THE CAN. JACK, I SEE YOU, SHE
ID. JACK WANTED TO GET OUT OF THE CAN. RUSS, HELP ME OUT, SAID JACK. I TH
K I AM STUCK. KIM WENT IN TO GET JACKS MOM AND DAD. JACK IS STUCK IN THE
N CAN, SHE SAID TO JACKS DAD. MOM AND DAD WENT OUT TO HELP. DAD BENT DOWN
ND HELPED JACK OUT OF THE CAN. THERE, YOU ARE OUT. HE SAID TO JACK, YOU M
T ALL PUT THE CUT GRASS BACK IN THE CAN, SAID DAD. CAN WE STILL PLAY. ASK
JACK. YES, SAID HIS MOM, BUT YOU MUST NOT PLAY IN THIS CAN. THEN JACK, R
S, AND KIM PUT BACK THE GRASS AND WENT BACK TO PLAYING.

*

FANTASY STORY #1

TRICKING WILL

PAN AN& THE DUCKS ARE PLAYING. WILL IS GOING OUT OF HIS DEN. HE SEES THE
CKS PLAYING. HE SEES PAN PLAYING WITH THEM. WILL IS BACK IN HIS DEN. HE
THINKING. I AM GOING TO GET THE DUCKS. I WILL TRICK THEM TO THE DEN. NEXT
I WILL GET PAN. WILL HAS A SACK. HE IS TRICKING THE DUCKS TO HIS DEN. WIL
PUTS THE DUCKS IN A TUB. THE DUCKS ARE YELLING. PAN CAN NOT SEE THEM. HE
NS TO THE DEN. I SEE DUCK TRACKS GOING IN THERE. PAN IS THINKING. I WILL
ELL FOR THEM. DUCKS. DUCKS. ARE YOU IN THERE. DUCKS. DUCKS. PAN KEEPS YEL
NG. BUT WILL HAS SHUT THE LID TO KEEP THE DUCKS FROM YELLING BACK TO HIM.
HEN PAN YELLS. WILL. WILL. ARE YOU IN THERE. WILL IS TRICKING ME. PAN IS
INKING. HE IS IN THERE. THE DUCKS MUST BE WITH HIM. I THINK THAT HE HAS R
THEM IN THAT TUB OF HIS. I MUST HELP THEM OUT. PAN YELLS IN TO WILL. YOU
N THERE. LEND ME THAT TUB. THE TUB WITH THE LID. I WANT TO PUT A HAM IN
A HAM. A HAM. WILL IS THINKING. I AM IN LUCK. HE TELLS THE DUCKS TO JUMP
UT. THEN HE SETS THE TUB DOWN NEXT TO PAN. THE DUCKS RUN WITH PAN TO A TR
. WILL IS IN BACK OF THEM. PAN PUTS ROCKS IN THE TUB AND SLAMS DOWN THE L
. WILL BENDS DOWN. PICKS UP THE TUB AND RUNS. I WILL KEEP THIS HAM. WILL
YELLING BACK TO THEM. HE THINKS THAT THERE IS A HAM IN HIS TUB. PAN TELL
THE DUCKS. TRICKING WILL IS FUN.

*

BEGIN PROCESSING

Attachment 4

Sample Run

BAS ANI STORY NUMBER 5

A PLAY

TIP. IN THIS PLAY, I AM ME. PAT. IN THIS PLAY, I AM ME. BUD. SEE THIS MASK ON ME. I AM A TREE. PAT. WE WILL CAMP NEXT TO THIS POND, TIP. TIP. YES, PAT . IT IS FUN TO CAMP OUT. TIP. HELP ME WITH THE TENT, PAT. WE NEED A LOG TO KEEP IT UP. PAT. SEE THE TREES. WE CAN GET ALL THE LOGS WE WANT. TIP. WE WI LL CUT THIS TREE FOR LOGS. TIP. PAT JUST WENT UP THE TREE TO CUT IT. TREE.

THIS WORD DOES NOT APPEAR IN THE BASIC WORD LIST OR IN THE BAS WORD LIST FOR

STOP. STOP. YOU MUST NOT CUT ME DOWN. PAT. DID THIS TREE ASK ME TO STOP. TR EE. YES, I DID. TIP. DO NOT STOP, PAT. WE NEED LOGS. TREE. I WILL TRICK THE M. I WILL JUMP UP AND DOWN. PAT. HELP. HELP. KEEP STILL, TREE. TIP. PAT FEL

THIS WORD DOES NOT APPEAR IN THE BASIC WORD LIST OR IN THE BAS WORD LIST DO

L FROM THE TREE. HE FELL IN THE POND. PAT. I AM ALL WET, TIP. HELP ME OUT. TIP. I CAN NOT SWIM, PAT. I CAN NOT HELP YOU. TREE. I WILL HELP YOU OUT, PA T. PAT. THE TREE HAS ME. TREE. I WILL PUT YOU ON THIS ROCK. PAT. I AM GLAD TO BE OUT OF THIS POND. I AM GLAD TO BE BACK ON LAND. TIP. TREE, WE WILL NO T CUT YOU DOWN.

A PLAY

TOTAL WORDS 208

UNIQUE WORDS 67

ALPHABETICAL DICTIONARY

	FREQ	CARD
A	2	2
ALL	2	4
AM	6	1
AND	1	8
ASK	1	6
BACK	1	12
BE	2	12
BUD	1	1
CAMP	2	2
CAN	3	4
CUT	4	5
DID	2	6
DOWN	3	6
FELL	2	8
FROM	1	9
FUN	1	3
GET	1	4
GLAD	2	11
HAS	1	11
HE	1	9
HELP	6	3
I	13	1
IN	3	1
IS	1	3
IT	3	3
JUMP	1	8
JUST	1	5
KEEP	2	4
LAND	1	12
LOG	1	3

LOGS	3	4
MASK	1	1
ME	8	1
MUST	1	6
NEED	2	3
NEXT	1	2
NOT	5	6
OF	1	12
ON	3	2
OUT	4	3
PAT	15	1
PLAY	2	1
POND	3	2
PUT	1	11
ROCK	1	11
SEE	2	1
STILL	1	8
STOP	4	6
SWIM	1	10
TENT	1	3
THE	7	3
THEM	1	7
THIS	8	1
TIP	11	1
TO	7	2
TREE	13	2
TREES	1	4
TRICK	1	7
UP	3	4
WANT	1	4
WE	7	2
WENT	1	5
WET	1	9
WILL	7	2
WITH	1	3
YES	2	2
YOU	5	6

FREQUENCY PER WORD

	FREQ	CARD
AND	1	8
ASK	1	6
BACK	1	12
BUD	1	1
FROM	1	9
FUN	1	3
GET	1	4
HAS	1	11
HE	1	9
IS	1	3
JUMP	1	8
JUST	1	5
LAND	1	12
LOG	1	3
MASK	1	1
MUST	1	6
NEXT	1	2
OF	1	12
PUT	1	11
ROCK	1	11
STILL	1	8
SWIM	1	10
TENT	1	3
THEM	1	7
TREES	1	4
TRICK	1	7
WANT	1	4
WENT	1	5
WET	1	9
WITH	1	3
SEE	2	1
ALL	2	4
GLAD	2	11
CAMP	2	2
DID	2	6
NEED	2	3
FELL	2	8
A	2	2
PLAY	2	1
BE	2	12
KEEP	2	4
YES	2	2
DOWN	3	6
IT	3	3
UP	3	4
CAN	3	4
LOGS	3	4
ON	3	2
IN	3	1

POND	3	2
STOP	4	6
CUT	4	5
OUT	4	3
NOT	5	6
YOU	5	6
AM	6	1
HELP	6	3
WILL	7	2
WE	7	2
THE	7	3
TO	7	2
ME	8	1
THIS	8	1
TIP	11	1
TREE	13	2
I	13	1
PAT	15	1

FICTION STORY NUMBER1

JACK GETS STUCK

KIM AND JACK SAT ON TOP OF A TRUNK. MOM AND DAD SAT NEXT TO THEM. LET US GO OUT AND PLAY, SAID JACK TO KIM. WE MUST ASK MOM AND DAD, SAID KIM. MOM, DAD, CAN WE GO OUT TO PLAY. ASKED KIM. YES, YOU CAN, SAID DAD. KIM AND JACK WENT OUT. PEGGY AND RUSS SAT NEXT TO A TREE. PLAY WITH US, SAID RUSS. PEGGY WANTED TO BE IT. RUSS, KIM, AND JACK ALL HID. RUSS HID IN BACK OF A TREE AND KIM HID IN BACK OF A BOX. JACK PICKED THE CUT GRASS OUT OF A CAN AND JUMPED IN. PEGGY WILL NOT SEE ME IN THIS CAN, JACK SAID. PEGGY SPOTTED RUSS IN THE GRASS NEXT TO THE TREE. RUSS, I SEE YOU, YELLED PEGGY. THEN PEGGY SPOTTED KIM IN BACK OF THE BOX. KIM, I SEE YOU, YELLED PEGGY. PEGGY DID NOT SEE JACK. RUSS AND KIM HELPED PEGGY. HE IS NOT IN BACK OF THIS LOG, SAID KIM. PEGGY JUMPED UP. I THINK I SEE HIM, SHE SAID. THE CUT GRASS IN OUT OF THAT CAN. I THINK HE HID IN THERE. PEGGY WENT TO THE CAN. JACK, I SEE YOU, SHE SAID. JACK WANTED TO GET OUT OF THE CAN. RUSS, HELP ME OUT, SAID JACK. I THINK I AM STUCK. KIM WENT IN TO GET JACKS MOM AND DAD. JACK IS STUCK IN THE TIN CAN, SHE SAID TO JACKS DAD. MOM AND DAD WENT OUT TO HELP. DAD BENT DOWN A

THIS WORD DOES NOT APPEAR IN THE BASIC WORD LIST OR IN THE FIC WORD LIST JACKS

ND HELPED JACK OUT OF THE CAN. THERE, YOU ARE OUT, HE SAID TO JACK. YOU MUS

THIS WORD DOES NOT APPEAR IN THE BASIC WORD LIST OR IN THE FIC WORD LIST JACKS

T ALL PUT THE CUT GRASS BACK IN THE CAN, SAID DAD. CAN WE STILL PLAY. ASKED JACK. YES, SAID HIS MOM, BUT YOU MUST NOT PLAY IN THIS CAN. THEN JACK, RUSS, AND KIM PUT BACK THE GRASS AND WENT BACK TO PLAYING.

TOTAL WORDS 309

UNIQUE WORDS 78

ALPHABETICAL DICTIONARY

	FREQ	CARD
A	5	1
ALL	2	5
AM	1	14
AND	15	1
ARE	1	16
ASK	1	2
ASKED	2	3
BACK	7	5
BE	1	5
BENT	1	15
BOX	2	6
BUT	1	18
CAN	12	3
CUT	3	6
DAD	9	1
DID	1	9
DOWN	1	15
GET	2	13
GO	2	1
GRASS	5	6
HE	3	10
HELP	2	13
HELPED	2	10
HID	4	5
HIM	1	11
HIS	1	18
I	8	8
IN	13	5
IS	2	10
IT	1	5

JACK	15	1
JUMPED	2	6
KIM	13	1
LET	1	1
LOG	1	10
ME	2	7
MOM	6	1
MUST	3	2
NEXT	3	1
NOT	4	7
OF	9	1
ON	1	1
OUT	10	2
PEGGY	11	4
PICKED	1	6
PLAY	5	2
PLAYING	1	19
PUT	2	17
RUSS	9	4
SAID	13	2
SAT	3	1
SEE	6	7
SHE	3	11
SPOTTED	2	7
STILL	1	17
STUCK	2	14
THAT	1	11
THE	12	6
THEM	1	1
THEN	2	8
THERE	2	12
THINK	3	11
THIS	3	7
TIN	1	14
TO	13	1
TOP	1	1
TREE	3	4
TRUNK	1	1
UP	1	11
US	2	1
WANTED	2	5
WE	3	2
WENT	5	3
WILL	1	7
WITH	1	4
YELLED	2	8
YES	2	3
YOU	7	3

FREQUENCY PER WORD

	FREQ	CARD
AM	1	14
ARE	1	16
ASK	1	2
BE	1	5
BENT	1	15
BUT	1	18
DID	1	9
DOWN	1	15
HIM	1	11
HIS	1	18
IT	1	5
LET	1	1
LOG	1	10
ON	1	1
PICKED	1	6
PLAYING	1	19
STILL	1	17
THAT	1	11
THEM	1	1
TIN	1	14
TOP	1	1
TRUNK	1	1
UP	1	11
WILL	1	7
WITH	1	4
ALL	2	5
PUT	2	17
SPOTTED	2	7
ASKED	2	3
STUCK	2	14
IS	2	10
BOX	2	6
THEN	2	8
THERE	2	12
JUMPED	2	6
GET	2	13
GO	2	1
ME	2	7
US	2	1
WANTED	2	5
HELP	2	13
HELPED	2	10
YELLED	2	8
YES	2	3
TREE	3	4
NEXT	3	1
SAT	3	1
SHE	3	11
CUT	3	6

WE	3	2
THINK	3	11
THIS	3	7
HE	3	10
MUST	3	2
NOT	4	7
HID	4	5
GRASS	5	6
A	5	1
PLAY	5	2
WENT	5	3
SEE	6	7
MOM	6	1
BACK	7	5
YOU	7	3
I	8	8
RUSS	9	4
OF	9	1
DAD	9	1
OUT	10	2
PEGGY	11	4
THE	12	6
CAN	12	3
IN	13	5
KIM	13	1
TO	13	1
SAID	13	2
JACK	15	1
AND	15	1

FANTASY STORY NUMBER2

WINK AND THE SHELL

WINK WANTS TO PLAY. HE IS GETTING OUT HIS DRUMS. HE IS PUTTING THEM UP NEXT TO A SHIP THAT HAS SUNK. WINK IS SITTING ON A ROCK. HE IS SITTING NEXT TO A SHELL. HE IS PLAYING THE DRUMS WITH STICKS. WINK WANTS TO REST. HE THINKS HE WILL NAP IN THE SHELL. HE IS PUTTING DOWN HIS SACK AND GETTING IN THE SHELL. WINK IS NAPPING ON THE SACK. THE SHELL IS SHUTTING. IT WILL LOCK HIM IN. WINK SEES THE SHELL SHUTTING. THIS IS A BAD SPOT TO BE NAPPING. I MUST GET OUT, HE YELLS. THE SHELL SHUT WITH WINK SITTING IN IT. HE IS SAD. WINK THINKS AND THINKS. THEN HE PICKS UP HIS SACK. HE DIGS DOWN IN THE SACK AND GETS OUT A PIN. WINK THINKS THE PIN WILL HELP HIM GET OUT OF THE SHELL. WINK IS SHUTTING THE SACK AND GETTING SET TO STICK THE PIN IN THE SHELL. WINK STUCK THE SHELL WITH THE PIN. STOP. STOP. SAID THE SHELL. LET ME OUT AND I WILL STOP, SAID WINK. JUST STOP STICKING ME WITH THAT PIN AND I WILL LET YOU GO, SAID THE SHELL. THE SHELL LETS WINK GO. WINK JUMPS OUT AND RUNS TO THE SHIP. I WILL REST ON THE SHIP, SAID WINK. WINK IS NAPPING ON THE SHIP. THE SHELL CAN NOT GET TO HIM THERE.

MINK AND THE SHELL

TOTAL WORDS 230

UNIQUE WORDS 75

ALPHABETICAL DICTIONARY

	FREQ	CARD
A	5	2
AND	7	4
BAD	1	6
BE	1	6
CAN	1	15
DIGS	1	8
DOWN	2	4
DRUMS	2	1
GET	3	7
GETS	1	9
GETTING	3	1
GO	2	13
HAS	1	2
HE	11	1
HELP	1	9
HIM	3	5
HIS	3	1
I	4	6
IN	6	4
IS	12	1
IT	2	5
JUMPS	1	13
JUST	1	12
LET	2	11
LETS	1	13
LOCK	1	5
ME	2	11
MUST	1	6
NAP	1	4
NAPPING	3	5

NEXT	2	1
NOT	1	15
OF	1	9
ON	4	2
OUT	6	1
PICKS	1	8
PIN	5	9
PLAYING	1	3
PLAY	1	1
PUTTING	2	1
REST	2	3
ROCK	1	2
RUNS	1	13
SACK	5	4
SAD	1	7
SAID	4	11
SEES	1	6
SET	1	10
SHELL	13	3
SHIP	4	2
SHUTTING	3	5
SHUT	1	7
SITTING	3	2
SPOT	1	6
STICKS	1	3
STICK	1	10
STICKING	1	12
STOP	4	11
STUCK	1	11
SUNK	1	2
THAT	2	2
THE	22	3
THEM	1	1
THEN	1	8
THERE	1	15
THINKS	4	3
THIS	1	6
TO	8	1
UP	2	1
WANTS	2	1
WILL	6	4
WINK	15	1
WITH	4	3
YELLS	1	7
YOU	1	12

	FREQ	CARD
BAD	1	6
BE	1	6
CAN	1	15
DIGS	1	8
GETS	1	9
HAS	1	2
HELP	1	9
JUMPS	1	13
JUST	1	12
LETS	1	13
LOCK	1	5
MUST	1	6
NAP	1	4
NOT	1	15
OF	1	9
PICKS	1	8
PLAYING	1	3
PLAY	1	1
ROCK	1	2
RUNS	1	13
SAD	1	7
SEES	1	6
SET	1	10
SHUT	1	7
SPOT	1	6
STICKS	1	3
STICK	1	10
STICKING	1	12
STUCK	1	11
SUNK	1	2
THEM	1	1
THEN	1	8
THERE	1	15
THIS	1	6
YELLS	1	7
YOU	1	12
DRUMS	2	1
NEXT	2	1
LET	2	11
THAT	2	2
DOWN	2	4
IT	2	5
ME	2	11
GO	2	13
UP	2	1
WANTS	2	1
PUTTING	2	1
REST	2	3
GET	3	7

HIM	3	5
HIS	3	1
SHUTTING	3	5
GETTING	3	1
SITTING	3	2
NAPPING	3	5
THINKS	4	3
SAID	4	11
I	4	6
ON	4	2
WITH	4	3
STOP	4	11
SHIP	4	2
SACK	5	4
A	5	2
PIN	5	9
IN	6	4
OUT	6	1
WILL	6	4
AND	7	4
TO	8	1
HE	11	1
IS	12	1
SHELL	13	3
WINK	15	1
THE	22	3

ALL INPUT PROCESSED